

Structured Preference Representation and Multiattribute Auctions

by

Yagil Engel

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2008

Doctoral Committee:

Professor Michael P. Wellman, Chair
Professor Edmund H. Durfee
Professor Jeffrey K. MacKie-Mason
Associate Professor Satinder Singh Baveja

© Yagil Engel

All Rights Reserved

2008

To Tamar, my beloved wife, and to our dear and lovely daughter, Daphna.

Acknowledgments

First and foremost I wish to thank my advisor, Michael Wellman, whose five years of help and guidance shaped my research. All but the last chapter of this work are the result of the collaboration with Mike, who accommodated my interest to explore topics in preference handling, despite being distinct from the directions pursued by the rest of the group. Mike was also instrumental in my transition from industry to academia. I met Mike when I joined TradingDynamics, a company for which he was working as a consultant. After collaborating with him for several years at work there was little doubt as for who would be my research advisor as a graduate student. Thanks to Mike, whose door is always (literally) open, I've had a smooth transition to becoming a student, after eight years as a software engineer.

My decision to pursue graduate studies, was influenced and encouraged by Yoav Shoham, the founder of TradingDynamics, for whom I also wish to express my deep gratitude. Yoav's influence during my years at work has been a major factor in this life decision, and his help has enabled me to carry it on.

The STIET program (Socio-Technical Infrastructure for Electronic Transactions), founded by Jeffrey MacKie-Mason and Michael Wellman, has had a major role in my studies. Beyond the financial help, it widened my research horizons with methods and current work in related disciplines, and helped me to maintain a broad perspective of electronic commerce research. In particular, I wish to thank Jeffrey MacKie-Mason, who was also on my dissertation committee, for his guidance and help throughout those years. I also thank my other committee members, Satinder Singh and Edmund Durfee.

Last but not least, I deeply thank Max Chickering, my mentor during an internship at Microsoft Live Labs, and with whom I have continued to collaborate thereafter. Chapter 7 is based on a joint work with Max, who is responsible for some of the results and edited parts of the text. A significant part of this work was done during the Live Labs internship, and the simulations were performed with the much appreciated help of Microsoft AdCenter.

Table of Contents

Dedication	ii
Acknowledgments	iii
List of Tables	viii
List of Figures	ix
List of Appendices	xi
List of Acronyms	xii
Abstract	xiv
Chapter 1 Introduction	1
1.1 Multiattribute Preferences	1
1.2 Preferences and Multiattribute Auctions	2
1.3 Thesis Contributions	5
1.4 Thesis Overview	7
Chapter 2 Background	9
2.1 Multiattribute Value and Utility	9
2.1.1 Preferences and Utility	9
2.1.2 Preferential Independence	12
2.1.3 Utility Independence	14
2.1.4 Additive Independence	16
2.2 Graphical Models of Preferences	19
2.3 Multiattribute Auctions	22
2.3.1 Basic Concepts	22
2.3.2 Characterizations	23
2.3.3 Buyer Revelation Mechanisms	24
2.3.4 Assuming Additive Preferences	26
2.3.5 Price Structures	27
2.3.6 Utility-Maximizing Mechanisms	30

2.3.7	Experimental Results	31
2.4	Strategic Sourcing	32
Chapter 3	Extensions to MAUT for the Trading Domain	34
3.1	Substitutes, Complements, and Utility Independence	34
3.2	Generalized Additivity Framework for MVF	37
3.2.1	Conditional Difference Independence	37
3.2.2	GAI Structure for MVF	38
3.3	Reasoning with Willingness-to-Pay	40
3.3.1	Construction	40
3.3.2	Optimization	42
3.3.3	An Example Scenario	42
3.3.4	Beyond Quasi-Linearity	44
Chapter 4	Structured Multiattribute Auctions	47
4.1	Introduction	47
4.2	GAI Auctions	50
4.2.1	GAI Preferences in Procurement	50
4.2.2	The GAI Auction	51
4.3	Analysis	53
4.3.1	Economic Properties	53
4.3.2	Computation and Complexity	58
4.3.3	Assumptions on Preferences	59
4.4	Example	60
4.5	Experimental Design	62
4.5.1	GAI Random Utility	63
4.5.2	Structured Subutility	64
4.5.3	Additive Approximation	65
4.6	Simulation Results	67
4.6.1	Measuring Efficiency	67
4.6.2	Efficiency and GAI Structure	68
4.6.3	Information Revelation	71
4.6.4	Computation and Complexity	72
4.6.5	Comparison for a Fixed Number of Rounds	74
4.7	Summary	74
Chapter 5	Multi-Unit Multiattribute Auctions	76
5.1	Introduction	76
5.2	MUMA preferences	77
5.2.1	The Global Multiattribute Allocation Problem	77
5.2.2	Preference Structures	79
5.2.3	<i>MMP-GMAP</i> decomposition	82
5.3	Aggregation Preferences	83
5.3.1	Aggregation over Trading Partners	83
5.3.2	Restricting Configuration Aggregation	85

5.4	MUMA GAI Auctions	86
5.5	Example	88
5.6	Two-Sided Multiattribute Call Market	90
5.7	Utility Representation and <i>MMP</i>	91
5.8	Solving <i>GMAP</i> under Allocation Constraints	93
5.8.1	Notation and Graphical Representation	94
5.8.2	Handling Indivisibility and Aggregation Constraints	94
5.8.3	Nonlinear Value of Quantity	99
5.8.4	Homogeneity Constraints	100
5.9	Experimental Results	102
5.10	Summary	105
Chapter 6	CUI Networks	107
6.1	Motivation	107
6.1.1	CUI Compared to other Independence Concepts	108
6.1.2	Graphical Models for CUI	110
6.2	CUI Networks	111
6.2.1	CUI DAG	111
6.2.2	CUI Decomposition	113
6.2.3	Properties of CUI Networks	116
6.3	CUI Modeling Example	117
6.4	Representation and Elicitation	121
6.4.1	Node Data Representation	121
6.4.2	Elicitation of Measurable Value Functions	121
6.5	Optimization	122
6.5.1	Optimization over CUI Trees	123
6.5.2	Optimization Over General DAGs	125
6.5.3	CUI Tree for Optimization of CAI Maps	127
6.6	Nested Representation	129
6.6.1	Discussion	130
6.6.2	Example	131
6.7	Summary	133
Chapter 7	Modeling Preferences for Sponsored Search	134
7.1	Introduction	134
7.2	Related work	136
7.3	Modeling User Experience	137
7.4	Publisher Utility Model	138
7.4.1	Publisher Value of User Utility	138
7.4.2	Probabilistic Model	139
7.4.3	Publisher Ranking Function	140
7.5	Sponsored Search Mechanism	141
7.5.1	Equilibrium Properties	142
7.5.2	Welfare Maximization	144
7.6	Model with Nonadditive Utility	145

7.6.1	Extending the Probabilistic Model	146
7.6.2	Utility Models	147
7.6.3	Mechanism for Nonadditive Utility	150
7.7	Short-Term Simulations	151
7.8	Long-Term Simulations	153
7.8.1	Click Model	154
7.8.2	Varying Tradeoff Knobs	156
7.8.3	Varying Utility Functions	158
7.9	Summary	159
Chapter 8	Conclusions	160
Appendices	164
Bibliography	186

List of Tables

Table

2.1	Previous literature on welfare maximizing multiattribute auctions.	30
4.1	GAI utility functions for the example domain.	60
4.2	Auction progression in phase A.	61
5.1	GAI utility functions for the multi-unit example domain.	88
5.2	Auction progression in phase A.	89
5.3	Mapping from combinations of allocation constraints to the solution methods of <i>GMAP</i>	102
6.1	Utility values for the Health and Wealth example (Bacchus and Grove, 1995).	109
6.2	Utility values for the Health, Wealth, and Location example.	109
6.3	Dependent and independent sets for each attribute.	118
6.4	Nested CUI decomposition.	132
B.1	Efficiency achieved by each auction type, given buyer and seller means, and variance of both.	185

List of Figures

Figure

2.1	CAI Map.	19
3.1	Map of CDI conditions.	43
4.1	HD procurement problem: (a) CDI map, (b) GAI network.	49
4.2	An example CDI map and GAI network.	51
4.3	Efficiency as a function of: (i) the size of largest GAI element (ξ), given $e = 5$, (ii) the number of GAI elements ($e + 1$), given $\xi = 5$	68
4.4	(i) Efficiency as a function of $k \geq 0$ (complements). (ii) Efficiency as a function of $k \leq 0$ (substitutes).	70
4.5	Efficiency of AP as a function of the number of sampling points used to devise the additive approximation.	70
4.6	Number of rounds as a function of: (i) the size of largest GAI element (ξ), given $e = 5$, (ii) the number of GAI elements ($e + 1$), given $\xi = 5$ and $\delta = 2$	73
4.7	Efficiency as a function of the number of rounds.	74
5.1	Multi-unit matching with two boolean attributes, A_1 and A_2	96
5.2	Generalized network flow model.	98
5.3	Extending the network flow model to express an XOR over quantities.	100
5.4	Alternative generalized network flow model.	103
5.5	Average performance of models when 30% of traders aggregate.	104
5.6	Average performance of models when 50% of traders aggregate.	104
5.7	Average performance of models when 70% of traders aggregate.	105
5.8	Performance of models when varying percentage of aggregating traders	105
6.1	CUI DAG given σ and order x_1, \dots, x_6	112
6.2	An example CUI network.	120
6.3	CUI networks in optimization examples: (i) Tree (ii) Non-tree	124
6.4	(i) A CAI map containing a cycle. (ii) Enhanced CAI map, expressing CUI of $\{a, d, f\}$ in b . (iii) An equivalent CUI tree.	128
6.5	Nesting example	132
7.1	Functional decomposition of $u(n_{cg}, n_{cb}, n_s)$	150

7.2	Revenue loss for improvement of relevance: (i) using LIN (ii) other utility functions, with $t_m = 0.4$	152
7.3	Revenue concession for: (i) demotion of bad ads, assuming all ads are shown, (ii) removal of bad ads.	153
7.4	Revenue each iteration, using various t_a, t_m , with different initial beliefs. . .	157
7.5	Total revenue for 2500 iterations, varying t_a and t_m	157
B.1	Efficiency of each auction as a function of: (i) the number of sellers (m), (ii) the size of the domain of each attribute (d).	184

List of Appendices

Appendix

A	Proofs	165
A.1	Chapter 3	165
A.1.1	Theorem 3.1	165
A.1.2	Theorem 3.2	165
A.1.3	Lemma 3.3 and Proposition 3.4	166
A.1.4	Theorem 3.6	167
A.1.5	Theorem 3.7	169
A.2	Chapter 4	170
A.2.1	Proofs of Section 4.3.1	170
A.2.2	Proposition 4.13	176
A.3	Chapter 5	177
A.3.1	Theorem 5.1	177
A.3.2	Theorem 5.2	178
A.3.3	Theorem 5.3	179
A.4	Chapter 6	180
A.4.1	Proposition 6.1	180
A.4.2	Lemma 6.3	180
A.4.3	Proposition 6.5	180
A.4.4	Proposition 6.6	180
A.4.5	Lemma 6.7	181
A.4.6	Lemma 6.8	181
A.4.7	Proposition 6.9	182
A.4.8	Proposition 6.10	182
A.4.9	Proposition 6.11	182
B	Additional Simulation Data	184

List of Acronyms

AD additive and discrete. 28

AI artificial intelligence. 1

AI additive independent. 16

AP additive approximation. 63

CAI conditionally additive independent. 17

CDI conditionally difference independent. 38

CPI conditionally preferential independent. 13

CUI conditionally utility independent. 15

DAG directed acyclic graph. 111

FOPI first-order preferential independence. 12

GAI generalized additive independent. 17

GMAP global multiattribute allocation problem. 79

HD hard drives. 47

IP integer programming. 97

MAP multiattribute allocation problem. 23

MAUT multiattribute utility theory. 9

MBR myopic best-response. 28

MMP multiattribute matching problem. 23

MPI mutually preferential independent. 13

MUI mutually utility independent. 15

MUMA multi-unit multiattribute. 76

MVF measurable value function. 11

NLD nonlinear and discrete. 27

PI preferential independent. 12

PK Parkes and Kalagnanam (2005). 27

RFQ request for quotes. 32

SB straightforward bidder. 54

UI utility independence. 14

VCG Vickrey-Clarke-Groves. 27

vNM von Neumann-Morgenstern. 10

WTP willingness-to-pay. 27

Abstract

Handling preferences over multiple objectives (or *attributes*) poses serious challenges to the development of automated solutions to complex decision problems. The number of decision outcomes grows exponentially with the number of attributes, and that makes elicitation, maintenance, and reasoning with preferences particularly complex. This problem can potentially be alleviated by using a factored representation of preferences based on independencies among the attributes. This work has two main components.

The first component focuses on development of graphical models for multiattribute preferences and utility functions. Graphical models take advantage of factored utility, and yield a compact representation for preferences. Specifically, I introduce CUI networks, a compact graphical representation of utility functions over multiple attributes. CUI networks model multiattribute utility functions using the well studied utility independence concept. I show how conditional utility independence leads to an effective functional decomposition that can be exhibited graphically, and how local conditional utility functions, depending on each node and its parents, can be used to calculate joint utility.

The second main component deals with the integration of preference structures and graphical models in trading mechanisms, and in particular in multiattribute auctions. I first develop multiattribute auctions that accommodate *generalized additive independent* (GAI) preferences. Previous multiattribute mechanisms generally either remain agnostic about traders' preference structures, or presume highly restrictive forms, such as full additivity. I present an approximately efficient iterative auction mechanism that maintains prices on potentially overlapping GAI clusters of attributes, thus decreasing elicitation and computation burden while allowing for expressive preference representation.

Further, I apply preference structures and preference-based constraints to simplify the particularly complex, but practically useful domain of multi-unit multiattribute auctions and exchanges. I generalize the iterative multiattribute mechanism to a subset of this domain, and investigate the problem of finding an optimal set of trades in multiattribute call markets, given restrictions on preference expression. Finally, I apply preference structures to simplify the modeling of user utility in sponsored-search auctions, in order to facilitate

ranking mechanisms that account for the user experience from advertisements. I provide short-term and long-term simulations showing the effect on search-engine revenues.

Chapter 1

Introduction

1.1 Multiattribute Preferences

The term *preferences* in the social sciences refers to the principle that guides choices made by a decision-making entity. In artificial intelligence (AI) we model preferences that guide the choices made by an artificial agent, which generally acts on behalf of a user or an organization. Preferences are most commonly represented quantitatively through a *utility function*. The concept of utility is ubiquitous in AI: it appears for example in the form of value of states in Markov Decision Processes, as a reward to learning agents, or payoff functions in game theory. In the general decision framework, based on the seminal work by von Neumann and Morgenstern (1944), probability distributions are used to weigh the utility values for each of the possible outcomes. Given a set of options, a rational agent chooses the one that maximizes its *expected utility*, that is the one that in expectation leads to the result it prefers the most, according to the preferences of the user or organization it acts for. Preferences therefore, have a significant role in decision making, similar to that of beliefs. Despite that, preferences and utilities have generally not received the level of attention AI researchers have devoted to beliefs and probabilities.

In this work I address preference handling, and the role of preferences in electronic trading. Most of the major components of this thesis focus on specific problems of automated trading, but the motivation and the main perspectives stem from the preference handling problem. More specifically, this work deals with *multiattribute* preferences. Decision outcomes are normally described by a set of features, or *attributes*, which can be seen as different objectives of the decision maker. Specification of preferences over the outcomes often requires reasoning about the *tradeoffs* among those attributes. The major challenge in preference handling is usually the *elicitation* of the preference information from users. The common approaches to practical preference elicitation (e.g., conjoint analysis, analytic hierarchy process (Saaty, 1980), pricing out (Clemen, 1996), and other elicitation techniques (Fishburn, 1967b)) is to assume that preferences are separable, or *additive*, meaning that

the value of one attribute does not affect preferences over the other. This is particularly the case in practical systems that require preference elicitation. Real preferences, in contrast, are rarely unconditional. Often the value of one attribute or objective depends on the availability or the value of another objective (this claim is supported by evidence from behavioral economics in reference to decision under uncertainty (Von Winterfeldt and Edwards, 1986)). The additivity assumption may seem superficially simplifying, while in fact the elicitation is less coherent when existing interdependencies are ignored. On the other hand, there are more subtle and more applicable independence properties of preferences, that can be employed to simplify elicitation and reasoning. Investigating and applying these structural properties is a central theme of this thesis.

Structural properties of preferences can help to reduce the dimensionality of the utility function, and thus achieve a compact representation of preferences. A compact representation streamlines computation tasks and communication, and most importantly reduces the elicitation burden. Moreover, preference structures can be treated as qualitative information that can often be reused over time, and sometimes across decision makers. It is therefore useful to identify and isolate this information, and thus the elicitation in each specific instance is limited to the quantitative information which is more variable. The qualitative structure can streamline the elicitation of the quantitative information, and also provide graphical insight into the problem. The combination of these goals is similar to the motivation behind *Bayesian Networks* (Pearl, 1988), and other graphical models for probability distributions, which have gained substantial academic and practical success. A main contribution of this thesis is the development and the application of graphical models for multiattribute utility functions.

1.2 Preferences and Multiattribute Auctions

A central goal in artificial intelligence is the automation of complex tasks, thus reducing labor costs, improving accuracy, and improving coordination between different entities. An important domain for automation in the Internet era is online trading. Several applications in this domain are well known, and incorporated for example in various shopping websites: auction, recommendation, and reputation systems, automated shopping software agents, decision support tools for product comparison, and more. Less familiar but arguably even more promising is the realm of business-to-business commerce. Significant portions of business of most companies occur in transactions with other companies, rather than consumers, creating an enormous potential market for online trading (Beil and Wein, 2003).

The choice of suppliers and contract negotiation in procurement is a realm that usually requires complex decision-making tasks, involving optimization problems, preferences of multiple stake holders, and strategic considerations.

Computer science research (and AI research in particular) can address electronic trading problems from several perspectives. Perhaps the two main perspectives are the following:

Optimization Algorithms for the selection of optimal allocations. In the procurement domain this amounts to the selection of suppliers and the exact goods to be procured.

Mechanism Design The extraction of information from the various participants (in procurement these are the procuring company and the candidate suppliers), while accounting for the incentives of these parties to reveal their information, and predict the outcome of the interaction using game-theoretic means (see Parkes (2001), Chapter 2).

Both tasks involve the preferences of the participants: we must consider the domain of possible outcomes, what are the preferences, or utilities of each party from each outcome, and strive to achieve optimal results based on that. We can therefore consider a third perspectives through which AI research should address problems from the trading domain: preference representation and elicitation, and application of preference structures such as those mentioned above.

To facilitate the introduction of multiattribute auctions, I illustrate this high-level classification in a more familiar research area: combinatorial auctions. In combinatorial auctions traders negotiate over combinations of goods, called *bundles*. Traders may bid on bundles out of a given set of goods, and this allows them to account for complementarity and substitutivity among the goods in the set. The computer science research in this field is vast, and summarized in a recent book (Cramton et al., 2006). The optimization problem here is to find an optimal allocation of the set of goods, given traders' valuations of the bundles. This is also called *the winner determination problem* and is addressed by, among others, Leyton-Brown et al. (2000), Sandholm (2002), and Sandholm et al. (2005). The combinatorial auctions mechanism design literature considers direct revelation mechanisms (MacKie-Mason and Varian, 1994) and iterative mechanisms (Wurman and Wellman, 2000; Parkes, 2001; Ausubel and Milgrom, 2002). Finally, researchers considered the preferences perspective, mainly through the eyes of bidding language design (Nisan, 2000; Boutilier and Hoos, 2001), through the interaction of preference expressiveness, optimization, and mechanism design (e.g., Lehmann et al. (2002) and Lehmann et al. (2006)), and through mechanism design that is based on a preference elicitation approach (Conen and Sandholm, 2000; Sandholm and Boutilier, 2006).

Combinatorial auctions are useful for various allocation problems, and in particular can be useful in procurement. In this work I focus on *multiattribute auctions*, which I consider an equally important component of procurement (see also Beil and Wein (2003)). Multiattribute auctions offer the ability to negotiate over various characteristics (attributes) of the good or service that is traded. An assignment of values to each of the attributes is called a *configuration*. In contrast to combinatorial auctions, in multiattribute auctions preferences are expressed over configurations of a specific good, rather than on bundles of different goods. We can consider traditional, price-only auctions as auctions in which the good is fully specified prior to the event. For example if a company wishes to procure pens through a price-only auction, it will have to fully specify the sizes of the pen, color, type of ink, and so on. In a multiattribute auction we can delay this commitment, and extract offers from suppliers that specify various cost structures for various values of these attributes. We can compare these offers with respect to the company's preferences over those criteria, and identify a configuration that maximizes the surplus of the deal. A utility-maximizing approach would seek to maximize the purchasing company's surplus (which is the difference between valuation and price), and a welfare-maximizing mechanism would consider the total trade surplus as the goal. Multiattribute auctions can therefore increase the surplus generated by the auction, because they extract multidimensional preferences from traders. Moreover, sometimes it is not at all possible to fully specify the procured good or service prior to the extraction of information from suppliers. In these cases multiattribute auctions are essential in order to facilitate a procurement event.

Multiattribute auctions have received far lower attention in the literature, in particular the computer science literature, than have combinatorial auctions (see survey in Section 2.3). In essence, both the combinatorial and multiattribute problems involve preferences over multidimensional domains, and the complexities of both are derived from the dimensionality. The combinatorial representation can be seen as a special case of the multiattribute one, where each attribute is restricted to be binary (i.e., whether the corresponding good is included or not). However, the allocation problem typically addressed by the two types of auctions is inherently different. A combinatorial auction seeks to find a combination of traders which together form the optimal assignment. A multiattribute auction considers preferences of both the auctioneer and the bidders, and seeks to find a bidder, and a configuration, which together yield the highest surplus with respect to the auctioneer's preferences.

Preference handling is a key issue in multiattribute auctions. A multiattribute mechanism must extract evaluative information over a complex domain of multidimensional configurations. Constructing and communicating a complete preference specification can

be a severe burden for even a moderate number of attributes, therefore practical multiattribute auctions must either accommodate partial specifications, or support compact expression of preferences assuming some simplified form. By far the most popular multiattribute form is the fully additive one, which is the simplest to represent and reason with. For example, several recent proposals for iterative multiattribute auctions (Beil and Wein, 2003; Bichler, 2001; David et al., 2002; Parkes and Kalagnanam, 2005) require additive preference representations.

Such additivity reduces the complexity of preference specification exponentially (compared to the general discrete case), but precludes expression of any interdependencies among the attributes. In practice, however, interdependencies among natural attributes are quite common. For example, the buyer may exhibit complementary preferences for size and access time (since the performance effect is more salient if much data is involved), or may view a strong warranty as a good substitute for high reliability ratings. Similarly, the seller's production characteristics (such as "increasing access time is harder for larger hard drives") can easily violate additivity. In such cases an additive value function may not be able to provide even a reasonable approximation of real preferences.

On the other hand, fully general models are intractable, and it is reasonable to expect multiattribute preferences to exhibit *some* structure. A major goal of this work, therefore, is to identify the subtler yet more widely applicable structured representations, and exploit these properties of preferences in trading mechanisms.

Therefore, there is a strong relationship between the preference handling perspective, and the other two mentioned above: optimization and mechanism design. In the next section I summarize the contributions of this thesis from each aspect.

1.3 Thesis Contributions

The contribution of this thesis is in the combination of preference handling theory with multidimensional trading. I suggest novel multiattribute preference handling approaches, and employ these and previously suggested tools to facilitate complex trading mechanisms and optimization algorithms.

The bulk of the contribution addresses these three perspectives above (preference handling, optimization, and mechanism design) simultaneously (Chapter 4, the first part of Chapter 5, and Chapter 7). I propose a novel efficient mechanism for multiattribute auctions that is based on an expressive modeling of preference structure. This mechanism is the first multiattribute auction mechanism that addresses the problem described above: it

accounts for essential interdependencies between attributes, but at the same time it takes advantage of the independencies that do exist, and facilitates tractable auctions over the multidimensional domain. Moreover, as an iterative, price-based mechanism it maintains the privacy of the buyer’s information, in contrast to most previous literature. In order to perform an experimental validation of the novel mechanism, I develop a framework to quantify the benefits of an accurate preference modeling in comparison to the additive representation.

I further carry the structured auctions idea on to a far more complex domain: that of multi-unit multiattribute auctions. I apply preference handling techniques to simplify this domain which has rarely been addressed in the mechanism design and auction literature before. This domain is nonetheless important for practical applications, and has been explored in practical contexts, but in general not with the economic rigor that I adhere to in this work. I propose the first mechanism design solution to a subset of this domain, relying on a novel preference representation scheme. The preference structures here are used not only to simplify the mechanism, but also to identify the subset of the domain to which it can be applied.

Another contribution addresses preference representation independently of the trading domain. I introduce a new graphical model for preferences, based on *conditional utility independence (CUI)*, a concept that has not been exploited for graphical models before . The motivation is part of a general claim I make in this thesis: by using weak independence concepts, that do not rely on additivity, we may be able to simplify complex decision domains, which cannot be factored using the stronger additive conditions. The graphical modeling encapsulates qualitative information about the preference order, leads to a compact representation of the utility function, and facilitates reasoning algorithms that I develop.

I also investigate the interaction between preference modeling and the generalized second price mechanism (GSP) implemented in sponsored-search auctions. Previous literature on GSP pursues efficiency results that ignore the role of the user in the process, and rarely consider the effect of sponsored search advertisements on the search user’s experience. I propose a principled approach to the incorporation of user utility in the mechanism, which requires modeling of the user’s utility from a set of advertisements on a search results page. The role of preference structures here is to simplify the model of the user’s utility over the space of possible sets of advertisements.

Another part of the work deals with optimization, ignoring mechanism design issues. It is also the only part of the work that deals with *two-sided* auctions, that is a market for multiple buyers and sellers. I consider the complexity of the global allocation problem in

multi-unit multiattribute call markets, as a function of restrictions made on expression of preferences. I provide a mapping between bid expressiveness and optimization complexity, and suggest optimization algorithms for several important cases.

Several additional contributions are in extending and adapting previous multiattribute theoretic results. Most notably, it facilitates the employment of recent AI-based research on preference structures, which is usually confined to the expected utility (vNM) framework, to the well-known framework of willingness-to-pay in economics.

1.4 Thesis Overview

After a detailed background on the major components of this work—multiattribute preferences, graphical models for preferences, and multiattribute auctions (Chapter 2)—I provide (Chapter 3) several insights and extensions that facilitate some of the subsequent developments. I formally introduce the notions of complements and substitutes in the multiattribute framework, notions that I find particularly useful in the realm of trading, and provide insights as to the relationship between these concepts and the notion of utility independence. Next, I lay out the utilitarian framework of the trading related work (Engel and Wellman, 2007), with several results that allow the application of particular preference structures in these settings. I provide an example that illustrates the conditional independence concept introduced there. Another such example, from the procurement domain, is provided in Chapter 4.

Chapters 4 and 5 focus on multiattribute auctions. Chapter 4 (based on Engel and Wellman (2007)) is the most direct example of the power of advanced preference handling for multidimensional trading: After describing the mechanism, I provide a detailed example, and show that the mechanism is approximately efficient under the assumption that the auction possesses an accurate representation of the buyer’s utility function. I also analyze computational properties. Next, I lay out a simulation framework that facilitates the comparison of a mechanism that captures the accurate preference structure of the buyer, with one that is restricted to additive preferences. The framework not only allows me to validate the advantages of the newly proposed mechanism, but may also serve as a guideline for future similar experiments whose goal is to quantify the benefits of an elaborated preference representation. I provide detailed simulation results regarding the efficiency, information properties, and computational complexity of GAI auctions, and compare the results with those obtained by an additive auction. The part of the chapter that deals with the simulations is based on Engel and Wellman (2008b).

Chapter 5 has two main components: in the first, based on Engel et al. (2007), I lay out a preference representation framework, and identify preference structures under which the mechanism of Chapter 4 can be generalized to this domain. The second part, based on Engel et al. (2006), deals with the computational problem of clearing a *two-sided* multiattribute call market. Based on main result of the first part of that chapter, I provide algorithms that simplify the clearing problem of several key special cases of bid expressiveness. I do that using intuitive and easy to implement network flow formulations, and provide suggestive performance results.

I return to preference representation, in a general sense, in Chapter 6. I introduce CUI networks, a graphical model for conditional utility independence. After comparison with previously suggested representations, I first show how the CUI condition can be expressed graphically by a directed acyclic graph, that supports a decomposition of the utility function to a lower dimensional representation. Next, I show how the utility function decomposes according to the CUI conditions, and can be represented compactly as local subutility functions at the nodes of the DAG. Subsequently, I consider elicitation procedures, optimization algorithms, and a graphical procedure that further reduces the dimensionality of the representation. This chapter is based on Engel and Wellman (2008a).

The last technical chapter (Chapter 7) studies the modeling of user utility in sponsored-search auctions. I construct a principled approach to this problem, through a utility-theoretic model of user's experience using expected utility, and the search engine publisher's problem of trading off user's utility and short-term expected revenue, and analyze the resulting mechanism. I use simulations to study the effects of the proposed ranking system and its parameters on short-term and long-term revenues (the first part of this chapter is based on Engel and Chickering (2008)).

Chapter 2

Background

In this chapter I provide the essential background, and cover previous literature relevant to the three main aspects of the work: multiattribute utility theory (MAUT), graphical models of preference and utility functions, and multiattribute auctions.

2.1 Multiattribute Value and Utility

I use standard terminology, concepts and notations from MAUT.¹ The definitions and results, unless noted explicitly, are adapted from the definitive text by Keeney and Raiffa (1976), and so are several results that are presented informally. As mentioned below, the results regarding generalized additivity are based on Fishburn (1967a) and the introduction to multiattribute measurable value functions is from Dyer and Sarin (1979). I present a unified view of multiattribute cardinal utility theory (mainly Section 2.1.3), which is a novel perspective compared to previous literature.

2.1.1 Preferences and Utility

Let Θ denote a space of possible states, which we refer to as decision *outcomes*. A decision maker's preferences are represented by a total pre-order, \preceq , over Θ . Preference orders considered in this work adhere to the economic notion of rationality, that is the order must be complete and transitive.

A rational preference order can be represented numerically, using a *value function*.

Definition 2.1. $v : \Theta \rightarrow \mathbb{R}$ is a value function representing \preceq if for any $\theta, \psi \in \Theta$, $v(\theta) \leq v(\psi)$ iff $\theta \preceq \psi$.

¹In some disciplines the term MAUT is used even when only the fully additive utility representation is referred to.

Clearly, any monotonic transformation of $v(\cdot)$ is also a value function for \preceq .

In many cases it is useful to represent, beyond a simple preference order over outcomes, a notion of *strength of preferences*. This requirement comes up when there exists some external factor that needs to be traded off with the strength of preferences over the outcomes. For example, when uncertainty is involved in the decision, strength of preferences over outcome must be weighed against their likelihood. In the *von Neumann-Morgenstern (vNM)* framework, uncertainty is represented using probability distributions. The decision maker does not choose a certain outcome, but rather an *action*, which results in some probability distribution over Θ . A distribution that results from an action is also called a *lottery*. Let \mathcal{P} denote the set of lotteries that result from the possible actions, so that any $p \in \mathcal{P}$ is a probability distribution $p : \Theta \rightarrow [0, 1]$. As each action results in a given unique lottery, we can consider the choice problem to be directly over \mathcal{P} .

In the vNM framework, we assume the existence of a preference order $\tilde{\preceq}$ over \mathcal{P} . A preference $p \tilde{\preceq} q$ (for some $p, q \in \mathcal{P}$) means that the decision maker, understanding the implication of each lottery, prefers that the distribution over the outcomes is p rather than q . For example, a person buying a lottery card for $\$x$ expresses preference towards the distribution of prizes promised by the card over the distribution that returns $\$x$ with probability 1.

Based on a set of axioms over $\tilde{\preceq}$ (Mas-Colell et al., 1995), the vNM framework establishes the existence of a value function that represents $\tilde{\preceq}$. As standard in the decision analysis literature, I use the term *utility function* in that framework.

Definition 2.2. A vNM utility function is a value function $\tilde{u} : \Theta \rightarrow \mathfrak{R}$, restricted as follows. For any $p \in \mathcal{P}$, let $\mathcal{E}_u(p) = \sum_{\theta \in \Theta} p(\theta) \tilde{u}(\theta)$.

Then for any $p, q \in \mathcal{P}$,

$$\mathcal{E}_u(p) \leq \mathcal{E}_u(q) \Leftrightarrow p \tilde{\preceq} q.$$

A monotonic transformation of $\tilde{U}(\cdot)$ preserves the ordinal preferences it represents (as it is the case for any value function), but does not necessarily preserves the preference order over lotteries. In other words, a monotonic transformation of $\tilde{u}(\cdot)$ still represents \preceq , but may or may not represent $\tilde{\preceq}$.

Definition 2.3. vNM utility functions $\tilde{u}_1(\cdot), \tilde{u}_2(\cdot)$ are called strategically equivalent if they represent the same preference order $\tilde{\preceq}$.

It can be inferred from the definition that any two vNM functions $\tilde{u}_1(\cdot), \tilde{u}_2(\cdot)$, are strategically equivalent if and only if there exist constants $b > 0, a$ such that

$$\tilde{u}_1(\cdot) = a + b\tilde{u}_2(\cdot).$$

That is, \preceq is preserved under a positive affine transformation, and if two vNM functions represent the same preference order then each must be a positive affine transformation of the other.

This uniqueness property characterizes any *cardinal utility function*. I skip the precise definition of the concept (Fishburn, 1976), but rather introduce another such function, one that will be used extensively in this work. This function is derived from measurement theory (Krantz et al., 1971), and it becomes useful in case the external factor is a measurement for which we can tradeoff the differences in the values of different outcomes. My terminology and definitions for this framework are based on Dyer and Sarin (1979).

In this framework we posit the existence of a preference order $\hat{\preceq}$ over *pairs of outcomes*. For some $\theta_1 \preceq \theta_2$ and $\psi_1 \preceq \psi_2$, the statement $(\theta_1, \theta_2) \hat{\preceq} (\psi_1, \psi_2)$ is interpreted to mean that the strength of preference of ψ_2 over ψ_1 is greater than or equal to the strength of preference of θ_2 over θ_1 . Krantz et al. (1971) establish the set of axioms ensuring the existence of a utility function representing $\hat{\preceq}$.

Definition 2.4. A measurable value function (MVF) is a value function $\hat{u} : \Theta \rightarrow \Re$, such that for any $\theta_1, \theta_2, \psi_1, \psi_2 \in \Theta$, for which $\theta_1 \preceq \theta_2$ and $\psi_1 \preceq \psi_2$, the following holds:

$$\hat{u}(\theta_2) - \hat{u}(\theta_1) \leq \hat{u}(\psi_2) - \hat{u}(\psi_1) \Leftrightarrow (\theta_1, \theta_2) \hat{\preceq} (\psi_1, \psi_2)$$

Note that an MVF can also be used as a value function representing \preceq , because $[\theta', \theta] \hat{\preceq} [\theta'', \theta]$ iff $\theta' \preceq \theta''$.

For a given decision problem of a specific decision maker, the MVF and the vNM functions are two distinct functions. Intuitively, the vNM function can be said to express both the preference difference expressed by the MVF, and in addition the *risk attitude* of the decision maker.² When the model involves uncertainty it is natural to apply utility under the vNM framework. When the outcome is modeled as certain, that is the decision maker directly selects an outcome from Θ , there is no point in modeling risk attitude. In fact, it can obfuscate the notion of strength of preferences we are interested in: if we elicit a vNM function from the decision maker, the difference in the utility values could reflect the decision maker's risk attitude.

I therefore employ MVF in the parts of this work that do not involve choice under uncertainty (Chapters 4 and 5). Other parts of the work refer to any cardinal utility representation, and I use the notation \preceq^* to denote any cardinal preference order and $u(\cdot)$ to denote any cardinal utility function.

²Dyer and Sarin (1982) perform an analytic comparison of the relationship between vNM functions and MVF.

In the multiattribute utility theory framework, an outcome in Θ is represented by a vector of values for n variables $A = \{a_1, \dots, a_n\}$, called *attributes* (I sometimes use other letters to denote attributes, but mention that explicitly). Each attribute $a \in A$ has a domain $D(a)$, so that $\Theta \subseteq \prod_{i=1}^n D(a_i)$. I usually use capital letters to denote subsets of attributes, small letters (with or without numeric subscripts) to denote specific attributes, and \bar{X} to denote the complement of X with respect to A . $D(X) = \prod_{a_i \in X} D(a_i)$ denotes the joint domain of the set of attributes X . An assignment to specific attributes, as well as joint instantiations of attribute subsets, is denoted by superscripts or prime signs. For example, $X^0 \in D(X)$ stands for the joint instantiation of a_i^0 for all $a_i \in X$. To represent an instantiation of subsets X and Y at the same time I use a sequence of instantiation symbols, as in $X'Y'$.

2.1.2 Preferential Independence

When A includes more than just a few attributes, reasoning over full outcomes is hard in several ways. Most notably, it is difficult for humans to compare outcomes over many dimensions, and complex for machines to store and analyze preferences over a number of outcomes that is exponential in the number of attributes. It is therefore useful to consider preference order over the joint product of some $X \subset A$, considering the rest of the attributes fixed on some predefined values. This is a *conditional preference order*. Such an order is also often referred to as a *ceteris paribus* preference order—one partial outcome is preferred to another *all else being equal*. I refer to Domshlak (2002) for a discussion of the validity of ceteris paribus preferences from a behavioral science point of view. Formally, it is defined as follows.

Definition 2.5. Outcome Y'' is conditionally preferred to outcome Y' given \bar{Y}' , if $Y'\bar{Y}' \preceq Y''\bar{Y}'$. The conditional preference order over Y given \bar{Y}' is denoted by $\preceq_{\bar{Y}'}$.

In general, conditional preferences may depend on the particular value chosen for the rest of the variables. More precisely, if $Y' \prec_{\bar{Y}'} Y''$, we can still find that $Y'' \prec_{\bar{Y}''} Y'$ for some $\bar{Y}'' \neq \bar{Y}'$. When this is the case, using the preference order $\preceq_{\bar{Y}'}$ may not provide any computational benefits. Fortunately, in many cases one can identify subsets Y for which this preference reversal does not occur, that is the preference order over Y is invariant to the value of \bar{Y} .

Definition 2.6. Y is preferential independent (PI) of \bar{Y} if $\preceq_{\bar{Y}'}$ does not depend on the value chosen for \bar{Y}' . This relationship is denoted by $PI(Y, \bar{Y})$.

Preferential independence can be very useful for qualitative preference assessment. *first-order preferential independence (FOPI)* (i.e., independence of a single attribute from

the rest) is a natural assumption in many domains. For example, in typical purchase decisions greater quantity or higher quality is more desirable regardless of the values of other attributes. Preferential independence of higher order, however, requires invariance of the *tradeoffs among some attributes* with respect to variation in others, a more stringent—though still often satisfiable—independence condition. The standard PI condition applies to a subset with respect to the full complement of remaining attributes. The *conditional* version of PI specifies independence with respect to a subset of the complement, holding the remaining attributes fixed.

Definition 2.7. Y is conditionally preferential independent (CPI) of X given Z ($Z = \overline{XY}$), if for any Z' , $\preceq_{X'Z'}$ does not depend on the value chosen for X' . This relationship is denoted by $CPI(Y, X \mid Z)$.³

The value function is defined over the joint domain. Therefore, a tabular representation of $v(\cdot)$ is exponential in $|A|$. Based on preferential independence, we can identify conditions that yield factored representations of the value function.

Definition 2.8. The set of attributes A are mutually preferential independent (MPI) if $\forall X \subset A, \quad PI(X, \overline{X})$.

The relationship between the MPI condition and the additive value function has been identified and proved in various ways. The earliest proof was provided by Debreu (1959).

Theorem 2.1. Assume $|A| \geq 3$. A is MPI iff there exists $\hat{v}(\cdot)$, a value function for \preceq , and a set of functions $v_i : D(a_i) \rightarrow \Re$ such that

$$\hat{v}(A) = \sum_{i=1}^n v_i(a_i).$$

This is called the *additive representation* and \hat{v} is called an *additive value function*. Equivalently, $v_i(a_i)$ can be replaced by *local value functions* $\bar{v}_i(a_i)$ that are normalized to $[0, 1]$, and multiplied by *scaling constants*, or *linear weights*:

$$\hat{v}(A) = \sum_{i=1}^n k_i \bar{v}_i(a_i).$$

³In all of the independence concepts notation, I allow the arguments to refer to either sets or single attributes. For example, $CPI(y, X \mid Z)$ is allowed as a shorthand for $CPI(\{y\}, X \mid Z)$.

2.1.3 Utility Independence

The MPI condition entails additivity of value functions, but is not sufficient for this convenient decomposition to hold for a cardinal utility function. Given MPI, there indeed exists a value function $\hat{v} = \phi \cdot u$ (for some monotonic function ϕ) that is additive, but ϕ is not necessarily linear, hence \hat{v} may not be strategically equivalent to $u(\cdot)$. In fact, a cardinal utility function requires its specialized independence concepts to achieve useful decompositions. I introduce the basic notions of independence for cardinal utilities, followed by the specialized extensions for the vNM and MVF frameworks.

The *conditional* cardinal preference order is defined similarly to the conditional preference order, and I use a similar notation, for example $\preceq_{\bar{Y}'}^*$ represents the conditional cardinal preference order over Y given the fixed value \bar{Y}' for \bar{Y} . Also note that conditional preference orders yield a functional representation in which the conditioning attributes are fixed. For example, the preference order $\hat{\preceq}_{\bar{Y}'}$ over lotteries on Y is represented by an MVF *conditional utility function*, $\tilde{u}(Y, \bar{Y}')$.

The counterpart of PI is called *utility independence*.

Definition 2.9. Y is utility independence (UI) of \bar{Y} , $UI(Y, \bar{Y})$, if the conditional cardinal preference order over Y , $\preceq_{\bar{Y}'}^*$, does not depend on value chosen for \bar{Y}' .

In the vNM framework, this condition requires the preference order over the lotteries in which the value of \bar{Y} is fixed on some \bar{Y}' , to be invariant to that fixed value. In the MVF framework, UI requires the *order over preference differences* over Y to be invariant. That means as follows: if given \bar{Y}' , the strength of which Y^2 is preferred over Y^1 is more than the preference strength of Y^4 over Y^3 , then this must also be the case given \bar{Y}'' .

If the conditional, cardinal preference order over Y does not depend on the value chosen for \bar{Y}' , it means that any function representing $\hat{\preceq}_{\bar{Y}'}$ will be strategically equivalent to a function representing $\hat{\preceq}_{\bar{Y}''}$ for any $\bar{Y}'' \in D(\bar{Y})$. Therefore, the conditional utility functions representing $\hat{\preceq}_{\bar{Y}'}$ and $\hat{\preceq}_{\bar{Y}''}$ must be positive affine transformations of one another. More precisely, there exist constants a and b such that,

$$u(Y, \bar{Y}'') = a + bu(Y, \bar{Y}').$$

This holds for any $\bar{Y}'' \in D(\bar{Y})$, with a potentially different constant for any such value. Hence in general we can represent $u(Y, \bar{Y}) = u(A)$ as follows:

$$u(A) = f(\bar{Y}) + g(\bar{Y})u(Y, \bar{Y}'), \quad g(\cdot) > 0. \quad (2.1)$$

This choice of \bar{Y}' is arbitrary, and can be replaced with any instantiation of $D(\bar{Y})$, resulting

in potentially different transformation functions $f(\bar{Y})$ and $g(\bar{Y})$. The utility function on the right hand-side of this equation is conditional. This implies that the UI condition leads to a functional decomposition of $u(\cdot)$: instead of one function of dimension $|Y| + |\bar{Y}|$, we need two functions of dimension $|Y|$ and one conditional utility function of dimension $|\bar{Y}|$. The functions $f(\cdot)$ and $g(\cdot)$ can also be represented in terms of conditional utility functions over Y .

UI can also be framed as a conditional relation.

Definition 2.10. Y is conditionally utility independent (CUI) of X given Z ($Z = \bar{X}\bar{Y}$) if for any Z' , $\preceq_{X'Z'}^*$ does not depend on the value chosen for X' . This relationship is denoted by $CUI(Y, X \mid Z)$.

CUI also supports functional decomposition. For any Z' , the conditional utility function over Y given $X'Z'$ is strategically equivalent to this function given a different instantiation of X . However, the transformation depends not only on X , but also on Z' . Hence we can write:

$$u(X, Y, Z) = f(X, Z) + g(X, Z)u(X', Y, Z), \quad g(\cdot) > 0. \quad (2.2)$$

Throughout the work I use the following notation, adapted from Fishburn (1967a). Let (a_1^0, \dots, a_n^0) be a predefined vector called the *reference outcome*. For any $X \subseteq A$, the function $u([X])$ stands for the projection of $u(A)$ to X where the rest of the attributes are fixed at their reference levels, meaning $u([X]) = u(X, X^0)$.

Continuing the analogy to ordinal preferences, the following is the global independence condition based on UI.

Definition 2.11. A is mutually utility independent (MUI) if every subset $X \subset A$ is UI of \bar{X} .

It may be natural to expect that MUI leads to an additive decomposition of a cardinal utility function, as does MPI for $v(\cdot)$. This is almost true—MUI may lead to either additive, or *multiplicative* decomposition, in which the utility function is a product of single-dimensional functions over the attributes. I provide the formal result, using the following notation: Let A^0 and A^1 be the least and most desired values, respectively, and $u(\cdot)$ be a utility function representing \preceq^* and normalized such that $u(A^0) = 0$ and $u(A^1) = 1$. Define $k_i = u([a_i^1])$, where the reference value is A^0 , and $u_i(a_i) = \frac{u([a_i])}{k_i}$. That is, $u_i(\cdot)$ is the conditional utility function over attribute a_i given the least desired values of the rest of the attributes, normalized to $[0, 1]$.

Definition 2.12. An MUI-factor of a set A of MUI attributes is a solution to

$$1 + k = \prod_{i=1}^n (1 + k k_i).$$

Keeney and Raiffa show that there is at most one MUI-factor in addition to zero (Appendix 6B of their text). This ensures the soundness of the following adaptation to their MUI representation theorem:⁴

Theorem 2.2. *Let A be a set of MUI attributes.*

1. *If the only MUI-factor of A is zero, then $u(A) = \sum_{i=1}^n k_i u_i(a_i)$.*
2. *Otherwise, let $k \neq 0$ be an MUI-factor. Then*

$$u(A) = \frac{\prod_{i=1}^n [k k_i u_i(a_i) + 1] - 1}{k}. \quad (2.3)$$

Keeney and Raiffa go on to point out that if $k > 0$ we can define $u'(A) = 1 + ku(A)$, a strategically equivalent function to $u(\cdot)$, and turn (2.3) into a multiplicative representation. This can be done in a similar fashion for $k < 0$. Further, they show that if MUI is known to exist, one elicitation query is sufficient in order to determine whether the form of the function is additive or multiplicative. In the next subsection I provide direct definitions for additive cardinal utility functions.

A weaker combination of UI conditions leads to the *multilinear* representation. If each X_i (possibly excluding one) is UI of its complement, U can be expressed as a multilinear combination of subutility functions on the X_i s. A drawback of the multilinear decomposition is that it requires specification of 2^n scaling constants. The multilinear structure is not used in this work.

2.1.4 Additive Independence

I initially limit the following discussion to the vNM framework.

Definition 2.13. *Let I_1, \dots, I_g be a partition of A . I_1, \dots, I_g are called additive independent (AI) if preferences over lotteries on I_1, \dots, I_g depend only on their marginal distributions over I_1, \dots, I_g .*

The following fundamental result relates this condition to an additive decomposition (the proof is available in (Fishburn, 1982), Section 11.1).

Theorem 2.3. *Let I_1, \dots, I_g be as above. Then I_1, \dots, I_g are additive independent if and only if there exist real-valued functions f_1, \dots, f_g such that*

$$\tilde{u}(a_1, \dots, a_n) = \sum_{i=1}^g f_i(I_i). \quad (2.4)$$

⁴The theorem is adapted from Keeney and Raiffa (1976), Theorem 6.1, page 289.

This decomposition can also be translated to use local, normalized utility functions coupled with linear weights.

Definition 2.13 is general, and a couple of special cases are worth mentioning. First, if $|I_r| = 1$ for all $r = 1, \dots, g$, then this is the additive decomposition that may result from MUI. Second, if $g = 2$ the condition is binary, similar to PI and UI. However, it is important to note that AI is fundamentally a symmetric conditions, unlike PI and UI. Moreover, it is straightforward to show that the binary AI(Y, \bar{Y}) entails UI(Y, \bar{Y}) and UI(\bar{Y}, Y), and each of these UI condition entails its PI counterpart.

Once we consider a binary AI, it is also useful to define its conditional version.

Definition 2.14. *Let X, Y, Z be a partition of the set of attributes A . X and Y are conditionally additive independent (CAI) given Z , if preferences over lotteries on A depend only on their marginal conditional probability distributions over X and Y . This relationship is denoted by $CAI(X, Y \mid Z)$*

This condition leads to a decomposition of the utility function to the form:

$$\tilde{u}(A) = \tilde{u}(X, Y^0, Z) + \tilde{u}(X^0, Y, Z) - \tilde{u}(X^0, Y^0, Z). \quad (2.5)$$

This is a decomposition of the utility function to two lower-dimensional functions, over the two overlapping subsets $X \cup Z$ and $Y \cup Z$. This provides greater flexibility compared to the notion of additivity over disjoint subsets. Due to the inherent symmetry of AI, the idea of a decomposition over potentially overlapping subsets can be generalized as follows.

Definition 2.15. *Let $I_1, \dots, I_g \subseteq A$ such that $\bigcup_{i=1}^g I_i = A$. I_1, \dots, I_g are called generalized additive independent (GAI) if preferences over lotteries on A depend only on their marginal distributions over I_1, \dots, I_g .*

What is now known as the GAI condition was originally introduced by Fishburn (1967a), and was named GAI and brought to the attention of artificial intelligence researchers by Bacchus and Grove (1995). Graphical models and elicitation procedures for GAI decomposable utility were developed under the vNM framework (Boutilier et al., 2001; Gonzales and Perny, 2004; Braziunas and Boutilier, 2005), and for a cardinal representation of the ordinal value function (Gonzales and Perny, 2005).

The following key result is proved by Fishburn (1967a).

Theorem 2.4. *Let $I_1, \dots, I_g \subseteq A$ such that $\bigcup_{i=1}^g I_i = A$. Then $I_1, \dots, I_g \subseteq A$ are GAI iff $\tilde{u}(\cdot)$*

decomposes as follows.

$$u(A) = \sum_{j=1}^g (-1)^j \sum_{1 \leq i_1 \leq \dots \leq i_j \leq g} \tilde{u}([\bigcap_{s=1}^j I_{i_s}]).$$

Note that (2.5) is a special case of this results. Further, the result implies that $\tilde{u}(\cdot)$ decomposes to lower-dimensional functions over the subsets I_r , for example:

$$\begin{aligned} f_1 &= u([I_1]), \text{ and} \\ \text{for } r = 2, \dots, g, \quad f_r &= u([I_r]) + \sum_{j=1}^{r-1} (-1)^j \sum_{1 \leq i_1 < \dots < i_j < r} u([\bigcap_{s=1}^j I_{i_s} \cap I_r]) \end{aligned} \quad (2.6)$$

Fishburn's proof relies directly on the definition of vNM utility. An interesting question arises whether this structure is similar for other cardinal utility function, notably for MVF. I provide a positive answer in Chapter 3. As background, I present previous work on additive structures for MVF, adapted from Dyer and Sarin (1979).

The basic notion is that of difference independence. The term $X \sim Y$ means that both $X \succeq Y$ and $Y \succeq X$.

Definition 2.16. *Attribute set $X \subset A$ is called difference independent of \bar{X} if for any two assignments $X^1 \bar{X}' \succeq X^2 \bar{X}'$,*

$$(X^1 \bar{X}', X^2 \bar{X}') \sim (X^1 \bar{X}'', X^2 \bar{X}'')$$

for any assignment \bar{X}'' .

In words, the preference differences on assignments to X given a fixed level of \bar{X} do not depend on the particular level chosen for \bar{X} .

Dyer and Sarin go on to define the conditions that achieve an analogous decomposition to that related to vNM from Theorem 2.3. Informally, given a partition I_1, \dots, I_g , the following three conditions are required: A is MPI, at least one of the sets I_i is difference independent of its complement, and A is *difference consistent*, a condition which ensures that any conditional preference order over differences is consistent with the corresponding conditional preference order over outcomes. A proof is provided by Dyer and Sarin (1977).

The concept of *reference value* mentioned above is worth a specific mention, as it is used in many parts of the work. The independence concepts described above reduce the dimensionality of the representation by employing conditional utility functions—conditioned on some predefined value. Though this value can be chosen arbitrarily, it is usually the most

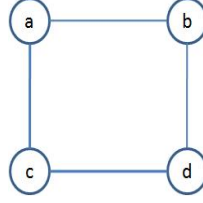


Figure 2.1: CAI Map.

convenient to pick the globally least desired value, usually denoted by A^0 (for which we normally assign $u(A^0) = 0$), and use its projection to specific attributes as their reference values. The notation $u([X])$ for any $X \subset A$ can then be used unambiguously instead of $u(X, \bar{X}^0)$. When referring to a reference value I do not in general restrict it to be the least desired one unless noted explicitly, but I do use this notation to with respect to an arbitrary reference value A^0 .

2.2 Graphical Models of Preferences

Perhaps the earliest effort to exploit separable preferences in a graphical model was the extension of influence diagrams by Tatman and Shachter (1990) to decompose value functions into sums and products of multiple value nodes. This structure provided computational advantages, enabling the use of dynamic programming techniques exploiting value separability.

Bacchus and Grove (1995) were first to develop a graphical model based on *conditional* independence structure. In particular, they establish that the CAI condition has a *perfect map* (Pearl and Paz, 1987); that is, a graph with attribute nodes A such that node separation reflects exactly the set of CAI conditions on A . More specifically, for any two sets of nodes $X, Y \subset A$, $\text{CAI}(X, Y \mid \bar{X}\bar{Y})$ holds if and only if there is no direct edge between a node in X and a node in Y . For example, the graph in Figure 2.1 represents the attribute set $A = \{a, b, c, d\}$, among which the two conditions $\text{CAI}(a, d \mid \{b, c\})$ and $\text{CAI}(b, c \mid \{a, d\})$ hold. I use the term *CAI map* of A when referring to the graph that reflects the perfect map of CAI conditions, in the context of a preference order over Θ . Bacchus and Grove go on to show that the utility function has a GAI decomposition over the set of maximal cliques of the CAI map. A clique is a set of nodes in which each pair is connected by an edge, and a maximal clique is a clique that is not contained within a larger one. For example, the maximal cliques of the graph in Figure 2.1 are $\{a, b\}$, $\{b, d\}$, $\{c, d\}$, $\{a, c\}$. Therefore, a utility function over A , defined on four attributes, can be represented using four two-dimensional

functions.

This result could seem to provide an alternative representation theorem to GAI, with a significant advantage: the GAI condition can be detected incrementally based on a set of CAI conditions. However, as noted by Bacchus and Grove, sometimes GAI conditions do not correspond to a collection of CAI conditions. They give the “triangle example”: the set of attributes $A = \{a, b, c\}$ may exhibit a GAI structure over the subsets $\{a, b\}, \{b, c\}, \{a, c\}$, without satisfying any conditional additive or utility independence condition. However, the possibility of taking advantage of such decompositions seems remote. No process to detect or verify the GAI condition directly has been proposed in the literature. CAI conditions, in contrast, are much simpler and more or less intuitive to detect, as they rule out any interaction between the independent subsets. It is sufficient to identify *pairs* of disjoint subsets that are independent, given that the rest of the attributes are fixed. Furthermore, procedures are available to verify CAI formally (Keeney and Raiffa, 1976). Therefore, it is much more conceivable to identify GAI structures that rise from the graphical model of CAI.

Initiating another important line of work, Boutilier et al. (2004a) introduced *CP networks*, an efficient representation for ordinal preferences over multiple attributes. In a CP network, each variable is conditionally PI of the rest given its parents. Ordinal multiattribute preference representation schemes (for decision making under certainty), and especially CP networks, can dramatically simplify the preference elicitation process, based as they are on intuitive relative preference statements that avoid magnitude considerations. CP networks received a considerable amount of study in subsequent literature. One extension, called *TCP networks* (Brafman and Domshlak, 2002; Domshlak, 2002), introduces a hierarchy of importance among the attributes, providing more expressive power. Furthermore, TCP nets allow the notion of importance between given attributes to be conditioned on value of other attributes.

Note that CP nets express only first-order dependencies. When, for example, FOPI holds for all attributes, a CP net is a disconnected graph that provides no information about the preference order. TCP nets extend CP nets to exhibit second-order dependencies. In practice, the dependencies may appear only at a higher order, meaning that the tradeoffs within a set of attributes are influenced by the value of a different attribute or attribute set. Moreover, this representation is limited to an ordinal preference order. In complex decision problems, tradeoff resolution may hinge in a complicated way on attribute settings over rich domains. This problem is particularly acute when continuous or almost continuous attributes are involved, such as money or time. In such cases, as well as in the presence of uncertainty, a cardinal preference order representation is required.

Boutilier et al. (2001) subsequently extended this approach to numeric, cardinal utility

in *UCP networks*, a graphical model that utilizes the GAI decomposition combined with a CP net topology. This requires dominance relations between parents and their children, somewhat limiting the applicability of the representation. The GAI structure was applied for graphical models in two other works: Gonzales and Perny (2004) employ the clique graph of the CAI map (the *GAI network*, see also Chapter 4) for elicitation purposes. Brafman et al. (2004) related the generalized additive form of an ordinal preference relation to a TCP-net structure.

Part of the motivation for graphical modeling of multiattribute utility is driven by the success of Bayesian networks in achieving compact representation and graphical reasoning algorithms for probabilities, based on conditional independence concepts. It is natural therefore that researchers explore the conceptual differences between probability and utility functions, and examine the opportunity to transfer results from one field to another. This direction was first advocated by Shoham (1997), who introduces a definition of a utility distribution function. His major claim is that the differences between the two representations (most notably, that probability is a set function, while utility can be defined only over points in the domain) is an arbitrary choice, and he goes on to define a notion of *utility distribution* on which one can apply a “Bayes rule” and use a Bayes-net like representation.

Later, La Mura and Shoham (1999) redefined utility independence as a symmetric multiplicative condition, taking it closer to its probability analog, and supporting a Bayes-net like representation. However, although multiplicative independence is different from additive independence, it is not necessarily weaker. Note that Utility Independence, as defined above, generalizes both notions. Furthermore, the authors claim that the multiplicative notion is more natural in preferences than the additive one, though this is not supported by decision-theoretic literature. I argue that in fact it is the UI condition that is more likely to hold and natural for decision makers to identify, as it expresses invariance of preference order.

Recent work by Abbas and Howard (2005) defines a subclass of utility functions on which a multiplicative notion of UI obeys an analog of Bayes’s rule.

A different and much earlier line of research was the introduction of utility trees by Gorman (1968), which were later further developed by Von Stengel (1988). The later work is the only graphical representation that uses UI as defined above. The utility tree relies on the notion of separability of subsets of the attribute set, to create a hierarchical breakdown that reduces the dimensionality of the representation to the dimension of the separable or utility independent sets. The application of utility trees to preference representation for planning was discussed by Wellman and Doyle (1992).

2.3 Multiattribute Auctions

The major application of this work on multiattribute preferences, is to *multiattribute auctions*.

Multiattribute trading mechanisms extend traditional, price-only mechanisms by facilitating negotiation over a set of predefined attributes representing various non-price aspects of the deal. Rather than negotiating over a fully defined good or service, a multiattribute mechanism delays commitment to specific configurations, in order to identify configurations that provide as high as possible surplus, under the preference representation and elicitation limitations of the mechanism. For example, a procurement department of a company may use a multiattribute auction to select a supplier of hard drives. Supplier offers may be evaluated not only over the price they offer, but also over various qualitative attributes such as volume, RPM, access time, latency, transfer rate, and so on. In addition, different suppliers may offer different contract conditions such as warranty, delivery time, and service. This example is developed in detail in Chapter 4.

2.3.1 Basic Concepts

The distinguishing feature of a multiattribute auction is that the goods are defined by vectors of *attributes*. I use the basic notations introduced in previous sections, that is A denotes a set of attributes describing the domain Θ . A *configuration* is a particular attribute vector, $\theta \in \Theta$. The outcome of the auction is a *set of bilateral trades*. Trade t takes the form $t = (\theta, q, b, s, p)$, signifying that agent b buys $q > 0$ units of configuration θ from seller s , for payment $p > 0$. For convenience, I use the notation θ_t to denote the configuration associated with trade t (and similarly for other elements of t). For a set of trades T , I denote by T_i that subset of T involving agent i (i.e., $b = i$ or $s = i$). \mathcal{T} denotes the set of all possible trades.

A *bid* expresses an agent's willingness to participate in trades. The semantics of a bid are specified in terms of *offer sets*. Let $\mathcal{O}_i^T \subseteq \mathcal{T}_i$ denote agent i 's *trade offer set*. Intuitively, this represents the trades in which i is willing to participate.

In general, an outcome of a multiattribute auction may involve a *set* of trades. Auctions that allow an agent to engage in multiple trades are called *multi-unit*, and those that restrict the outcome to a single trade per agent are called *single-unit*. The trade in single-unit multiattribute auctions may include a quantity, but in that case the quantity can be framed as an attribute in A . Initially I limit the discussion to single-unit auctions, and introduce the multi-unit framework in Chapter 5. In the single-unit model I focus on *procurement*,

the common application of multiattribute auctions. In the procurement model there is a single buyer, who has a utility function $u_b(\theta)$ for purchasing $\theta \in \Theta$.⁵ There are m sellers s_1, \dots, s_m with cost functions $c_i(\cdot)$, representing the cost for s_i to supply configurations in Θ to the buyer. The following definitions are adapted from Parkes and Kalagnanam (2005).

Definition 2.17. *The multiattribute allocation problem (MAP) is defined as:*

$$\max_{i \in \{1, \dots, m\}, \theta \in \Theta} u_b(\theta) - c_i(\theta). \quad (2.7)$$

An allocation (s_i^*, θ^*) selected above is said to maximize the *social welfare* of the procurement problem.

MAP can be decomposed to two subproblems: first find the most efficient configuration for each trader, and then find the trader whose efficient configuration yields the highest surplus. We call the first part as the *multiattribute matching problem* (Engel et al., 2006).

Definition 2.18. *The multiattribute matching problem (MMP) for a buyer b and a seller s_i is defined as:*

$$\text{MMP}(b, s_i) = \arg \max_{\theta \in \Theta} u_b(\theta) - c_i(\theta).$$

I call a configuration θ^* selected by $\text{MMP}(b, s_i)$ a *bilaterally efficient* configuration for s_i . Let $t = (\theta_t, q, b, s_j, p)$ represent an outcome of the auction. The *buyer's profit* is defined as $u_b(\theta_t) - p$, and the *seller's profit* of s_j is $p - c_j(\theta_t)$. In many multiattribute procurement auction designs, the term *scoring rule* refers to the function $r : \Theta \rightarrow [0, 1]$ used by the auctioneer to evaluate suppliers' bids. It can be regarded as the buyer's report of his utility function, that is for a truthful buyer $r(\cdot)$ is strategically equivalent to $u_b(\cdot)$.

2.3.2 Characterizations

I structure previous literature over the characterizations which I consider most crucial in the design of multiattribute auctions.

Expressiveness The expressiveness of preferences, through bids. Previous literature can roughly be divided according to whether or not preference representation is limited to additive utility functions.

Economic Properties Whether the proposed mechanism provides economic guarantees, under well-specified assumptions. A well-known goal is to maximize total welfare,

⁵In Section 3.3 I formalize the specific utility function used here.

or surplus, of the trade (welfare maximization) (Myerson and Satterthwaite, 1983). Another common goal in price-only auctions is to maximize the auctioneer's (expected) revenue (Myerson, 1981). The multiattribute generalization of this concept is utility-maximization (Beil and Wein, 2003), where the utility is the value of the allocation net of payment.

Tractability Whether the auction can be practically executed for large domains. This criterion refers to overall execution time, including the computation time required from participants.

Preference Revelation Information revealed to sellers by the auction. The main issue is the degree to which the auction reveals the auctioneer's (buyer's) utility function prior to and during bidding.

Price Feedback *Iterative mechanisms* can guide bidders to the efficient parts of the configuration space by providing interim information that summarizes previous bids. This is in contrast to direct revelation mechanisms, which require sellers to produce their complete cost functions.

As explained below, the criterion of preference revelation by the auctioneer is a crucial characterization of multiattribute literature. I therefore first consider papers that require the full revelation of buyer's utility prior to bidding. Also, I initially restrict the discussion to welfare-maximizing mechanisms, and consider utility maximization in Section 2.3.6.

2.3.3 Buyer Revelation Mechanisms

An early economic analysis of multiattribute auctions is due to Thiel (1988). In Thiel's model, the procurer's budget is fixed, and he does not value price reduction—a particularly unrealistic assumption in procurement. The full scoring rule is revealed at the beginning of the auction. The bidders bid vectors of attribute values, but since all parties know that these bids are evaluated using $r(\cdot)$, this is equivalent to bidding the values $r(\cdot)$ directly. He shows that some basic properties of pricing theory are retained, and the model therefore reduces to a familiar single-dimensional environment.

Che (1993) analyzes two-dimensional (price and quality) procurement auctions, modeled after some US government process. In Che's model, the attributes domain Θ represents the quality of the good. The buyer reveals a scoring rule $r(\cdot)$, and evaluates bids according to $\hat{s}(\theta, p) = r(\theta) - p$ where $\theta \in \Theta$ and p denotes the price. A seller s_i has a private cost function $c_i(\cdot)$ over Θ .

Under several assumptions, most notably that supplier costs are not correlated, he suggests three types of auctions. In all of the three, bidders bid price and quality values.

First-score. Direct generalization of the first-price sealed-bid auction. The highest score bidder wins and must supply the exact quality and price he committed to.

Second-score. The highest score bidder wins, and must provide a combination of price and quality that achieves the second-best score.

Second-preferred offer. The highest score bidder wins and must supply the quality and price offered by the second-best bidder.

Che shows the following basic result, which essentially reduces the bidding to a single dimension.

Lemma 2.5. *In any equilibrium of first-score and second-score auctions, seller s_i offers the quality $\theta_i = \arg \max_{\theta' \in \Theta} (r(\theta') - c_i(\theta'))$.*

Note that $r(\cdot)$ represents the buyer's utility function as reported, hence θ_i above is in fact the solution to $MMP(b, s_i)$.

Che goes on to show as a result that the first-score auction is efficient under the unique symmetric equilibrium of the game, which is a generalization of the first-price equilibrium in single-dimensional auctions. In the second-score mechanism, the equilibrium in dominant strategies is truthful bidding. A bidder s_i has only one truthful strategy: report θ_i as above, and price $c_i(\theta_i)$. These results generalize their single-dimensional counterparts, and provide a theoretical basis for subsequent work on multiattribute auctions. In the second-preferred offer protocol the actual quality and price values mean nothing to the bidder, and the result above does not hold. Therefore, any combination of price and quality that achieve the highest score that the bidder is capable of (meaning the highest score with nonnegative profit), is a (symmetric) Bayes-Nash equilibrium. This protocol does not seem to provide economic benefits in comparison to the other two.

Che's results are the basis of what I see as the most popular approach in the multiattribute procurement auction literature: The buyer reveals a full utility function to the sellers. The sellers are then expected to find a configuration that maximizes the difference between that utility and their own cost, and bid only on that configuration. This approach practically reducing the problem to single-dimensional bidding. The downside is of course the information revelation burden it imposes on the buyer. There are two sources to this concern. First, Che shows that the buyer has incentives to drive the quality evaluation below the truthful value, potentially leading to inefficiency. Second, the buyer may be reluctant to reveal this information for strategic reasons, as elaborated in Section 2.3.5.

Che's basic assumption of non-correlated costs is not applicable in practice—supplier costs are much more likely to be correlated than not. The question is therefore how do the results change when this assumption does not hold. This is one of the key topics explored by Branco (1997). Branco generalizes Che's model to correlated costs (efficiencies). A firm's efficiency depends on other firms' efficiencies, and are not fully known to the firm before the types of the other firms are revealed. Therefore, in contrast to Che's model, firms cannot bid optimal qualities in a single stage. Branco suggests a first stage for type revelation, and a second one in which a bilateral bargaining procedure is conducted over the quality to be provided. Since the full efficiency parameters are now known, optimal quality can be calculated and the buyer can make a take-it-or-leave-it offer with the optimal quality. The seller has incentive to accept that offer. Branco analyzes the equilibrium behavior of firms for a first-score or second-score version of this mechanism, and finds that in both cases the expected payment is optimal. Therefore, this generalizes Che's revenue equivalence result to correlated costs.

This foundational work does not explore any potential structure in the quality space, hence does not address an issue that is key in practical and computational application of multiattribute auctions: Preference representation over multidimensional domains.

A later work that requires buyer revelation and also remains agnostic to preference structures is by Vulkan and Jennings (2000). These authors suggest a modified version of English auctions (iterative auctions that require new bids to increment over current bid price) under which bidders are required to improve current score, rather than price. This is an adaptation of Che's second-score auction to an iterative mechanism. In contrast to this work, most later literature assumes that preferences are additive, and I survey some of that literature below.

2.3.4 Assuming Additive Preferences

When considering multidimensional quality over a large set of attributes, the expression of full utility functions over the domain becomes intractable, unless the representation is restricted. As I mention in the introduction, most previous proposals take the extreme assumption of full additivity. Bichler (2001) argues that an additive scoring function almost always applicable, and there is no real need to model more complex preferences. In contrast, I believe that behavioral studies that claim to support the additive representation are applicable only to the ordinal value function. I elaborate on that distinction in Chapter 3.

David et al. (2002) extend Che's model by introducing m non-price attributes rather than one. The authors assume both additive preferences, and in addition that the sellers'

utility functions differ over a single cost parameter. Further, the mechanism requires full revelation from the buyer's side. Additivity is assumed in two more recent papers: I discuss the work by Beil and Wein (2003) under utility maximizing mechanisms in Section 2.3.6, and the one by Parkes and Kalagnanam (2005) in the Section 2.3.5.

In essence, the additive forms used in trading mechanisms assume MPI over the set of attributes $A \cup \{p\}$, where p is the monetary payment. Intuitively that means that *willingness-to-pay (WTP)* for value of an attribute or attributes cannot be affected by the value of other attributes. I formalize this notion in the next chapter.

2.3.5 Price Structures

The literature surveyed above emphasizes that auctions require the buyer to reveal a full utility function prior to bidding. This is a major obstacle to practical adaptation of these mechanisms. As argued by Rothkopf et al. (1990), auctions are rarely an isolated event, and information obtained regarding valuation of a trader by some other traders can be used later to his disadvantage. This is particularly true in procurement, where the buyer-supplier relationships evolve and change over time, and suppliers may retain significant market power in subsequent interactions. Furthermore, events are sometimes conducted on a recurrent basis, and several events may be conducted for related goods with correlated valuations. In addition, the buyer may wish to keep secret the way her utility may be discriminating for or against particular suppliers (Koppius, 2002).

One approach that avoids full revelation was suggested by Parkes and Kalagnanam (2005) (PK). PK first state the sell-side Vickrey-Clarke-Groves (VCG) mechanism that solves *MAP*. Assuming all traders reveal full utility functions, the winning seller gets his VCG payment. The VCG payment is the amount that ensures that the seller's profit is equal to his marginal product, which is the difference between the value of *MAP* with and without that seller. This scheme ensures that it is every seller's best interest to reveal his true utility function (i.e., the mechanism is *incentive compatible* for sellers), but it is not necessarily the buyer's best interest to report her accurate utility.

PK go on to introduce to the *nonlinear and discrete (NLD)* mechanism, which is an iterative mechanism: a descending-price auction for general preferences. In NLD, the buyer reports her full utility function to the auction system—not to the sellers. A price is defined for each configuration in Θ . In every round, sellers submit bids on configurations of their choice, and the price is reduced on any configuration that received a seller bid except for the one designated as the provisional allocation. The provisional allocation is selected to maximize the buyer's profit with respect to current prices.

PK adapt the notion of *myopic best-response (MBR)* to this context: sellers are myopic best-responders if they bid optimally under the assumption that current prices are final. PK's MBR definition requires sellers to bid on *all* optimal configurations. If all sellers follow this strategy, the auction leads to a competitive equilibrium with maximal prices (which are also the VCG prices). This allows PK to show that this strategy profile is an ex-post Nash Equilibrium and the outcome is efficient (in the limit, when the bid increment is taken to zero). An ex-post Nash Equilibrium means that agents cannot improve their outcome by changing their strategy unilaterally, *regardless of the costs of other sellers*, and regardless of the reported utility function of the buyer.

NLD, however, is not tractable when there are more than a few attributes, because of the exponential price space it needs to maintain and the potentially exponential number of configurations that MBR bidders are expected to bid on. This leads the authors to suggest auction *additive and discrete (AD)*. AD requires that all traders have additive preferences. Prices are maintained for every value of each attribute (rather than on the joint domain) and sellers place bids for levels of each attribute, rather than for configurations. The sellers also bid a special discount value, so the price of a configuration from a seller is the sum of the prices of the attribute level minus the discount suggested by the seller. The discount allows surplus from one attribute to be transferred to other attributes. In other words, the competition is per attribute level (in order to drive prices down), but at some point sellers might get into a deadlock where a bidder currently winning on one attribute cannot beat the winner of another attribute. In such case they are required to provide a discount. The competition over the discount helps to select seller that can provide the maximal sum of surpluses over the attributes.

The economic properties of AD are analyzed using a primal-dual formulation, over a linear program that solves MAP. Because the linear program maximizes surplus, the competitive equilibrium (CE) supports the efficient outcome. The auction protocol maintains the (ϵ) complementary slackness condition and leads to the (maximal) CE. Maximal CE prices correspond to VCG payments for the sellers.

The authors subsequently study the extent to which information is revealed by the buyer and sellers in the auction, through simulation analysis. They model the amount of information that a trader manages to keep as the subspace of feasible weights of attributes in his utility function. They show how prices of configurations translate to constraints over weights under the additive representation. They conduct the experiment for both auction types, but the simulation framework assumes that preferences are additive also when experimenting with NLD. It is not clear how this might have affected the results.

Whereas auction NLD requires sellers to provide their exact weights, auction AD allows

sellers to retain 30-50% of their private information.

It is important to note that the mechanisms suggested by PK do not provide strong incentive guarantees on the buyer side. The buyer can achieve higher surplus by reporting an untruthful utility function. This must be considered with respect to the well known impossibility result by Myerson and Satterthwaite (1983). Roughly, this states that no mechanism exists that is efficient (i.e., achieves maximal surplus in an equilibrium) when monetary transfer is restricted to be between the agents, and no agent expects to lose from participating in the mechanism. Given that, relaxing the buyer's incentives is arguably a reasonable compromise. In order to increase her profit, the buyer needs to change the utility function such that the difference between the top two sellers is reduced. To do that she must predict the bidding of sellers and guess the configurations each would ultimately land on. It is hard to see how this is possible without knowing sellers' cost functions.

These two auction protocols are extended by Sunderam and Parkes (2003), to incorporate proxy bidding. The proxy is a bidder agent, which collects information about the preferences of the bidder and submits bids on her behalf. A single-dimensional version is the simple mechanism employed by eBay, encouraging a bidder to submit a stronger bid than the minimum currently required to win (up to his true willingness-to-pay) but increasing the winning bid only with the minimum increment until reaching the value given by the bidder. The multiattribute version is of course more interesting, and can potentially save more bidding effort. The paper focuses on methods by which the proxy learns the bidder's weights, using the information measurement approach discussed above. Each choice the bidder makes puts a constraint on the relevant weights. When a proxy needs to submit a specific bid in order to become the provisional winner, it needs to verify that the required bid is implied by the current set of constraints, and only when it is not the proxy needs to elicit more information from the bidder. The potential of proxies in MA auctions is intriguing, and invites further research.

Another relevant work is due to Shachat and Swarthout (2002). Their EBC auction is a two phase mechanism: in the first phase the buyer assigns bidding credit to sellers according to the quality of the good they offer, and in the second the sellers compete in an English auction which is effectively conducted over the *surplus* they provide to the buyer. This method also avoids the buyer revelation issue, but compromises surplus: the space of bilateral matches between the buyer and each of the seller is not explored (i.e., the trading configuration might not constitute a solution to *MMP*), hence the mechanism potentially misses configurations that yield higher surplus than the configurations suggested by the bidders.

To summarize this part, Table 2.1 provides an outlined view of previous multiattribute

	Full Buyer Revelation	Incremental Buyer Revelation
Additive Preferences	Bichler (2001) David et al. (2002)	AD (Parkes and K., 2005)
General Preferences	Che (1993); Branco (1997) Vulkan and Jennings (2000)	NLD (Parkes and K., 2005)

Table 2.1: Previous literature on welfare maximizing multiattribute auctions.

literature. Considering the design criteria above, I limit the discussion to mechanisms that provide welfare-maximization (or approximate maximization) guarantees (under specific assumptions), and map previous literature according to the information revelation and preference expressiveness criteria. Most of the mechanisms above are designed to promote competition. Some of them (Che, 1993; Branco, 1997) provide abstract results that can lead to iterative implementations. A key motivation for Chapter 4 is to design a mechanism for the most attractive quadrant of Table 2.1 (general preferences and incremental revelation), that can be *tractable* for a large set of attributes.

2.3.6 Utility-Maximizing Mechanisms

Another common design objective for economic mechanisms is to maximize the auctioneer’s own utility, rather than the total welfare. Note that the maximization is defined with respect to the space of all possible *mechanisms* and under some notion of equilibrium behavior. A mechanism that is expected to achieve maximal utility in that sense is called an optimal auction. The only work I am aware of that directly addresses the problem of optimal multiattribute auction design is due to Ronen and Lehmann (2005).

These authors propose a simple two-stage VCG-based mechanism. The idea is that if agents reveal their cost functions, we can choose the k agents whose cost functions generate the highest potential surplus (over all configurations) and drop the others. The second stage is a VCG mechanism in which the remaining agents participate, with a reserve utility that is equal to the potential surplus of the best dropped agent. The reserve utility ensures that no dropped agent would have had incentive to misrepresent its type in order to make it to the next stage. The second stage VCG mechanism can be made optimal (under the reserve utility constraint) based on the distribution over the cost functions of the surviving agents (ignoring their previous reports), conditioned on the reported cost functions of the dropped agents. The authors prove that the expected utility of the proposed mechanism is at least half of the expected utility of the optimal one.

Beil and Wein (2003) suggest an iterative mechanism for buyer's utility maximization in procurement. Under the assumption of truthful sellers, the buyer conducts iterative rounds in order to extract information about the sellers' cost functions, and uses this information to design a scoring function that extracts maximal surplus from the winning seller.

My approach follows PK, under the belief that welfare-maximizing mechanisms are more beneficial in the long-term. They promote trust between the buyer and his trading partners, and encourage sellers to participate. In PK's work and in mine the buyer is assumed to bid truthfully, and the mechanism ensures incentive compatibility to the sellers. This is justified by a similar argument, relying on the buyer's interest in maintaining healthy long-term relationship with the chosen suppliers.

2.3.7 Experimental Results

A few papers report results of lab experiments on multiattribute auctions. Bichler (2000) examined the core questions of multiattribute utility theory in a relatively realistic environment. The auction is conducted by an electronic broker for OTC financial derivatives. Buyers are invited to start auctions and provide their additive utility function—a rare experiment in extracting scoring functions from users. Buyers first enters weights (called importance values) for each of the attributes. They are asked to provide *attribute value functions* (single dimensional function on each attribute), either as ratio values for discrete attributes, or linear functions for continuous variables. Under the assumption that the scoring function represents the buyer's utility function, the multiattribute auctions, not surprisingly, achieved significantly higher utility values than price-only auctions. Next, the author compared first-score and second-score auctions. The result is that revenue equivalence (or, more precisely, the *utility equivalence*) is not achieved—first score achieves higher utility values (for the buyer) than an English auction. Even more surprisingly, the English auction achieves higher value than a direct revelation second-score mechanism (*Sealed Bid*). However, these results were repeated (and even aggravated) in the single-dimensional auction experiment, therefore it does not seem to characterize multiattribute auctions in particular.

Strecker and Seifert (2004) compare two second-score mechanisms—one iterative and the other direct revelation. The iterative mechanism is an English auction, where sell bids are compared based on the buyer's scoring function. The buyer's full scoring function is revealed to suppliers up front in the two experiments, and the suppliers know their exact cost schedule. As in Bichler's experiments, English auctions achieve higher utility for the buyer, and with it a larger surplus, than does second-score direct revelation.

Here too, the interesting question that is not addressed is to what extent this result can be attributed to the fact that the auctions are multiattribute. It might indeed be the case that the additional difficulty of calculating their optimal bids under the second-score format causes bidders to deviate from their dominant strategy. However, it is important to note that the subjects of this experiments were inexperienced. In practice, I expect suppliers who participate in serious procurement events to employ experienced decision makers with sensible bidding methodology.

In a different experiment (Strecker, 2003), two kinds of English auctions are compared: one in which buyer reveals full utility function (TF) and one in which the buyer does not reveal the utility function (TN). The results is that TF led to more efficient outcomes. Koppius (2002) performed experiments in which subjects bid in a procurement event of a chemical. The chemical had two attributes in addition to price. He also finds that revealing the buyer's utility function (coupled with price feedback) increases the efficiency of multiattribute auctions.

These results are quite expected, as can be predicted by Che's analysis, because sellers that know the buyer's preferences can optimize their own bidding to those points that yield higher surplus, in order to increase their chances to win. The approach advocated by PK, whose spirit I follow in Chapter 4, avoids this tradeoff between efficiency and the buyer's privacy of information, and allows to retain both through the use of a multiattribute price structure.

2.4 Strategic Sourcing

The term *strategic sourcing* refers to the repeated process in which a company identifies goods or services for which it wishes to engage in a new procurement contract, selects a supplier or suppliers, and implements contractual engagement. The common practice for the selection of suppliers is, very roughly, as follows. The procurement department of the company sends out a *request for quotes (RFQ)* to a set of candidate suppliers. Through the RFQ the department collects prices offered by suppliers, and additional information that includes characteristics of the good each supplier offers, such as quality measurements, performance, and so on, other aspects of the deal such as warranty and payment terms, and information about the supplier. Various stakeholders within the company, which may for example represent departments that have interest in the decision, consider the suppliers' responses, bring in additional information they might have about each supplier, and together choose the supplier that best answers everybody's needs. In some cases the selection pro-

cess is divided to two stages: a *Request for Information* stage, in which suppliers provide information about their companies, after which the procurement department sends an RFQ to a selected subset of the responders. In most cases, the process takes several additional iterations involving sequential or simultaneous bilateral negotiation with particular suppliers (Beall et al., 2003).

A *sourcing software* can automate or partially automate this lengthy process, and can yield various advantages to the company (Beall et al., 2003; Sandholm, 2007). In particular, automation of the RFQ can provide the following benefits:

- Reduce the cost of the event by various means, in particular by automating, or at least aiding the decision making process using preferences elicited from the various stake holders and other sources of data.
- Increase social welfare of the outcome, by using advanced auction technologies (e.g. multiattribute auctions), and by accommodating participation from a larger number of suppliers without a significant increase in the cost of the event.
- Increase the buyer's share of the surplus, by creating more competitive pressure through iterative auctions.

In this work I focus on the multiattribute component of the problem: as described above, the information provided by a supplier usually contains values to numerous attributes of the good, the contract, and the supplier. For example, in 2001 KLM Royal Dutch Airlines performed a (single) sourcing event to select a supplier of buttercups for its in-flight service. This apparently simple procurement item required 77 attributes!⁶

Despite the potential benefits described above, procurement auction softwares have not been widely adopted in the industry (Beall et al., 2003). When auctions are used, most commonly these are price-only auctions (Bichler et al., 2006), which as I argue in Section 2.3, and as argued by Sandholm (2007), may yield inferior efficiency. This slow adoption could be a result of several challenges in the development of such software, which have not been fully addressed in practice or in academic literature. First, the elicitation of preferences from various stake-holders is extremely challenging in such high-dimensional domains. Furthermore, as described in Section 2.3, there are difficulties in designing multiattribute auctions, in particular the design of tractable, efficient, iterative auctions, that do not compromise the privacy of the buyer's preference information. In this work I attempt to provide answers to some of these challenges.

⁶Source: <http://www.negometrix.com>

Chapter 3

Extensions to MAUT for the Trading Domain

In this chapter I present several developments that extend previous results in multiattribute utility theory. These results, though mostly straightforward to show, are necessary in order to support some of the work in the next chapters. They are either motivated by the trading domain (Section 3.1) or required to adapt MAUT results for that domain (Sections 3.2 and 3.3).

3.1 Substitutes, Complements, and Utility Independence

Utility independence, as introduced by Keeney (1971), provides a way to reduce the dimensionality of preference representation, under a reasonable and intuitive independence assumption. One of the motivations to use UI is the substantial previous work in eliciting and employing this condition (Keeney and Raiffa, 1976). In this section I examine intuitive properties of the UI and CUI conditions, and additional means to establish their existence.

The UI decomposition (2.1) employs two transformation functions, $f(\bar{Y})$ and $g(\bar{Y})$. Keeney and Raiffa (1976) show that these functions can be represented using conditional utility functions over \bar{Y} . I first adapt this result to CUI. Assume $\text{CUI}(Y, X \mid Z)$ holds. Then we can determine $f(\cdot)$ and $g(\cdot)$ of (2.2) by solving the system of two equations below, applying the specific instantiations Y^1 and Y^2 :

$$\begin{aligned}u(X, Y^1, Z) &= f(X, Z) + g(X, Z)u(X^0, Y^1, Z), \\u(X, Y^2, Z) &= f(X, Z) + g(X, Z)u(X^0, Y^2, Z),\end{aligned}$$

yielding

$$g(X, Z) = \frac{u(X, Y^2, Z) - u(X, Y^1, Z)}{u(X^0, Y^2, Z) - u(X^0, Y^1, Z)}, \quad (3.1)$$

$$f(X, Z) = u(X, Y^1, Z) - g(X, Z)u(X^0, Y^1, Z). \quad (3.2)$$

Note that Y^1, Y^2 are selected such that $Y^1 \prec_{X^0} Y^2$ (this relation does not depend on Z), meaning $u(X^0, Y^2, Z) - u(X^0, Y^1, Z) \neq 0$.¹

Whereas additivity practically excludes any interaction between utility of one attribute or subset (X in Definition 2.14) to the value of another (Y), utility independence admits common notions of dependency. These types of dependency are known as *complements* and *substitutes*. These concepts are known primarily in the context of combinatorial preferences, that is preferences over combinations of distinct items. In the multiattribute framework, these relationships can be intuitively described as follows. Two (sets of) attributes are complements if an improvement in the value of both is worth more than the sum of the same improvement of each independently, whereas the two attributes are substitutes if it is the other way round. Clearly, these concepts are meaningful only with respect to (sets of) attributes that are preferentially independent, otherwise the notion of “improvement” is conditional on the value of other attributes.

As argued earlier, typical purchase and sale decisions exhibit first-order preferential independence. In other types of decisions this condition is also rather common, as it occurs whenever an attribute has some natural order of quality. Therefore, it is intuitively easier to initially limit the definition to a relationship between single attributes.

Definition 3.1. Let $\hat{u}(\cdot)$ be a measurable value function over S' . Let $a, b \in S'$, and $Z = S' \setminus \{a, b\}$, and assume that a and b are each FOPI of the rest of the attributes. Attributes a and b are called *complements* if for any $a^i > \hat{a}$ ($a^i, \hat{a} \in D(a)$) and any $b^j > \hat{b}$ ($b^j, \hat{b} \in D(b)$), and any $Z' \in D(Z)$,

$$\hat{u}(a^i, b^j, Z') - \hat{u}(\hat{a}, \hat{b}, Z') > \hat{u}(a^i, \hat{b}, Z') - \hat{u}(\hat{a}, b^j, Z') + \hat{u}(\hat{a}, b^j, Z') - \hat{u}(\hat{a}, \hat{b}, Z').$$

Attributes a and b are *substitutes* if the inequality sign is (always) the other way round.

Now consider two subsets of attributes, $X, Y \subset A$, each of which is PI of its complement. Definition 3.1 can easily be extended to such subsets: X and Y are complements, for example, if the preference differences between joint values of X are magnified given

¹In the rare case that such Y^1 and Y^2 do not exist, a different X^0 can be selected. If for any X^0 , indifference holds for any two values of Y , the attributes of Y are not essential for the decision problem.

a more desirable joint value of Y . If all of those differences are magnified such that the *order* over them does not change, X may still be UI of Y . Note also that if Y and X are complements (substitutes) then any pair $x \in X$ and $y \in Y$ are complements (substitutes).

Next I establish a connection between Definition 3.1 and factors that are used in the UI driven decompositions.

Theorem 3.1. *Let A be a set of attributes, and X, Y, Z a partition of A , such that $CUI(Y, X | Z)$. Let $g(X)$ be the corresponding factor in the decomposition (2.2). Then*

1. *Y and X are complements iff $g(X) > 1$, for any choice of X^0 .*
2. *Y and X are substitutes iff $g(X) < 1$, for any choice of X^0 .*

The proof (in Appendix A) relies on the formulation (3.2) above. Intuitively, this equation reflects the ratio between improving the value of Y given X , and the same improvement of Y given $X^0 \preceq X$ (Z is fixed for all outcomes). If the ratio is greater than one, it means that the first improvement is worth more than the second. The formal definition characterizing the case of $g(X) = 1$ is provided in the next section. As explained in the next chapter, the result can help in preference elicitation of compact utility functions.

Theorem 3.2. *Let A be a set of MUI attributes, such that there is a MUI-factor $k \neq 0$. Then $k > 0$ iff all pairs of attributes in A are complements, and $k < 0$ iff all pairs of attributes in A are substitutes.²*

This result can provide intuitive help in determining the MUI-factor. In addition, it can be useful for experimental purposes. Suppose that a modeler wishes to simulate, with random data, an application that operates based on users' preferences. In order for the random data to imitate utility of real users, the modeler might like to consider sets of attributes which are complements or substitutes. This provides a single control parameter (the MUI-factor) to determine the strength of complementarity or substitutivity, and allows one to express simulation results as a function of that parameter. I do exactly that in Chapter 4.

Attributes often can be shown to be complements or substitutes intuitively (see examples in this chapter and in Chapter 4). However, even when such meaningful dependencies do not hold, there are more subtle relations that rule out additive independence. In the vNM framework, one such condition is the so-called *multiattribute risk aversion*. Von Winterfeldt and Edwards (1986) bring the following example: consider a choice between two even-chance lotteries, α and β . By choosing α , we either win a fancy TV, or win a fancy Wi-Fi system. With β , we either win both or nothing. The two lotteries yield the same marginal probability, 0.5, over each of the two attributes ("win fancy TV" and "win fancy

²An intuition for this result, limited to the specific case of MUI between two attributes, is also given by Keeney and Raiffa (1976).

Wi-Fi”). However, a risk-averse decision maker naturally prefers lottery α , to ensure that he wins at least one of the prizes, and therefore the attributes are not additively independent. Von Winterfeldt and Edwards (1986) cite behavioral economics experimental results, supporting the claim that the additive representation is usually a poor representation of preferences under uncertainty. The authors refer to multiattribute risk aversion as a major explanation for that phenomena.

I will now argue that this example can be adapted to illustrate that additivity is too restrictive under the MVF framework as well. The reasoning is as follows: under uncertainty, nonlinearity of a single-dimensional utility function is often justified as risk aversion (or *risk proneness*). When no uncertainty is involved, single-dimensional utility function may still be nonlinear, if the decision maker exhibits *decreasing* (or increasing) *marginal utility*. Similarly, the notion of multiattribute risk aversion has an analogous notion which can be referred to as *decreasing multiattribute marginal utility*. Consider the following example: a car buyer is asked for the amount he is willing to pay for having a sun roof in the car, the amount he is willing to pay for an upgrade of the audio system, and the amount he is willing to pay for both. If the decision maker have decreasing margins over fancy additions to the car—he is reluctant to add two large amounts at the same time for something that is not a necessity—the latter amount would be lower than the sum of the other two, meaning that the two attribute are not *difference independent*, again ruling out an additive form. The two attributes in such case are substitutes according to the definition above, even though there is nothing inherent in one that substitutes for the other.

3.2 Generalized Additivity Framework for MVF

Since the CAI and GAI conditions are defined based on preferences over lotteries, we cannot apply Bacchus and Grove’s result (Section 2.2) and Fishburn’s (Theorem 2.4) to MVF without first establishing an alternative framework based on preference differences. I develop such a framework, ultimately producing a GAI decomposition (Eq. 3.5) of the MVF.

3.2.1 Conditional Difference Independence

My first step is to generalize Definition 2.16 to conditional independence.

Definition 3.2. Let X, Y, Z be a partition of the set of attributes A . X is conditionally

difference independent (CDI) of Y given Z , denoted as $CDI(X, Y \mid Z)$, if

$$\forall \hat{Z} \in D(Z), X^1, X^2 \in D(X), Y^1, Y^2 \in D(Y), \quad (X^1 Y^1 \hat{Z}, X^2 Y^1 \hat{Z}) \sim (X^1 Y^2 \hat{Z}, X^2 Y^2 \hat{Z}).$$

The conditional set is always the complement of the independent sets (here $Z = A \setminus (X \cup Y)$), hence I sometimes leave it implicit, using the abbreviated notation $CDI(X, Y)$ (and similarly for the CUI and CPI conditions).

CDI leads to a decomposition similar to that obtained from CAI (Eq. 2.5).

Lemma 3.3. *Let $\hat{u}(A)$ be an MVF representing preference differences. Then $CDI(X, Y \mid Z)$ iff*

$$\hat{u}(A) = \hat{u}(X^0, Y, Z) + \hat{u}(X, Y^0, Z) - \hat{u}(X^0, Y^0, Z).$$

A convenient generalization of Lemma 3.3 is as follows.

Corollary 3.4. $CDI(X, Y \mid Z)$ iff there exist functions $\psi_1(X, Z)$ and $\psi_2(Y, Z)$, such that

$$\hat{u}(X, Y, Z) = \psi_1(X, Z) + \psi_2(Y, Z). \quad (3.3)$$

An immediate consequence of Corollary 3.4 is that CDI is a symmetric relation. Furthermore, Theorem 3.1 can be extended as follows: $CDI(X, Y \mid Z)$ holds iff $g(X) = 1$, for any choice of X^0 .

The conditional independence condition is much more applicable than the unconditional one. For example, if attributes $a \in X$ and $b \notin X$ are complements or substitutes, X cannot be difference independent of \bar{X} . However, $X \setminus \{a\}$ may still be CDI of \bar{X} given a .

3.2.2 GAI Structure for MVF

A single CDI condition decomposes the value function into two parts. I seek a finer-grain global decomposition of the utility function, and in particular it seems plausible that MVF supports a GAI condition, analogous to Definition 2.15. To this purpose I now employ the results of Bacchus and Grove (1995), who establish that the CAI condition has a *perfect map* (see Section 2.2). Their proofs can easily be adapted to CDI, since they rely only on the decomposition property of CAI that is also implied by CDI according to Proposition 3.4. I refer to this CDI equivalent of a CAI map as a *CDI map*.

Theorem 3.5. *Let $G = (A, E)$ be a perfect map for the CDI conditions on A (the CDI map). Then*

$$\hat{u}(A) = \sum_{r=1}^g f_r(I_r), \quad (3.4)$$

where I_1, \dots, I_g are (overlapping) subsets of A , each corresponding to a maximal clique of G .

Given Theorem 3.5, we can now identify an MVF GAI structure from a collection of CDI conditions. The CDI conditions, in turn, are particularly intuitive to detect when the preference differences carry a direct interpretation, as in the case with monetary differences discussed below. Moreover, the assumption or detection of CDI conditions can be performed incrementally, until the MVF is decomposed to a reasonable dimension. This is in contrast with the fully additive decomposition of MVF that requires mutual preferential independence (Dyer and Sarin, 1979).

Theorem 3.5 defines a decomposition structure, but to represent the actual MVF we need to specify the functions over the cliques. The next theorem establishes that the functional constituents of MVF are the same as those for GAI decompositions as defined by Fishburn (1967a) for vNM (Theorem 2.4).

Theorem 3.6. *Let $G = (A, E)$ be a perfect map for the CDI condition on A , and $\{I_1, \dots, I_g\}$ a set of maximal cliques as defined in Theorem 3.5. Then the elements of functional decomposition (3.4) can be defined as*

$$\begin{aligned} f_1 &= \hat{u}([I_1]), \text{ and} \\ \text{for } r = 2, \dots, g, \quad f_r &= \hat{u}([I_r]) + \sum_{j=1}^{r-1} (-1)^j \sum_{1 \leq i_1 < \dots < i_j < r} \hat{u}([\bigcap_{s=1}^j I_{i_s} \cap I_r]) \end{aligned} \quad (3.5)$$

The proof (in Appendix A) directly shows that if graph $G = (A, E)$ is a perfect CDI map, $\hat{u}(A)$ decomposes to a sum over the functions defined in (3.5). Thus this proof does not rely on the decomposition result of Theorem 3.5, only on the existence of the perfect map.

The result is a GAI form which is analogous to a vNM GAI condition that is constructed from a collection of CAI conditions. The analogy is therefore not complete—as mentioned in Section 2.2, in vNM a GAI decomposition may exist without any related CAI condition. However, as also discussed in that section, such a GAI condition is rarely useful.

To summarize, the results of this section generalize additive MVF theory. In particular it justifies the application of methods recently developed under the vNM framework (Bacchus and Grove, 1995; Boutilier et al., 2001; Gonzales and Perny, 2004; Braziunas and Boutilier, 2005) to representation of cardinal value under *certainty*.

3.3 Reasoning with Willingness-to-Pay

Trading decisions represent a special case of decisions under certainty, where choices involve multiattribute outcomes and corresponding monetary payments. In such problems, the key decision often hinges on relative valuations of price differences compared to differences in alternative configurations of goods and services. Theoretically, price can be treated as just another attribute, however, such an approach fails to exploit the special character of the money dimension, and can significantly add to complexity due to the inherent continuity and typical wide range of possible monetary outcome values.

As explained in Section 2.1.1, we might opt for an MVF representation when we can point out specific attributes that can be used as a relative preference measurement for others. A natural choice for trading is to use monetary scaling, where the preference difference over a pair of outcomes represents the difference in willingness to pay (WTP) for each.

3.3.1 Construction

In this section I apply measurable value to represent differences of willingness to pay for outcomes. I assume that the agent has a preference order over an outcome space. The outcome space is represented by a set of attributes A , and an attribute p representing monetary consequence. Note that in evaluating a purchase decision, p would correspond to the agent's money holdings net of the transaction (i.e., wealth after purchase), not the purchase price. An outcome in this space is represented for example by (θ', p') , where $\theta' \in \Theta$ is an instantiation of A and p' is a value of p . I use the notation \succeq to denote a preference order over the domain $\Theta \times D(p)$. I further assume that preferences are *quasi-linear* in p (Mas-Colell et al., 1995), that is there exists a value function of the form $v(A, p) = \bar{u}(A) + L(p)$, where L is a positive linear function.³

The quasi-linear form immediately qualifies money as a measure of preference differences, and establishes a monetary scale for $\bar{u}(A)$.

Definition 3.3. Let $v(A, p) = \bar{u}(A) + L(p)$ represent \succeq , where p is the attribute representing money. $\bar{u} : \Theta \rightarrow \Re$ is called a willingness-to-pay (WTP) function.

Note that WTP may also refer to the seller's "willingness to accept" function. Since

$$(\theta_1, p') \succeq (\theta_2, p'') \Leftrightarrow \bar{u}(\theta_1) - \bar{u}(\theta_2) \leq L(p'' - p')$$

³In many procurement applications, the deals in question are small relative to the enterprises involved, so the quasi-linearity assumption is warranted. This assumption can be relaxed to a condition called *corresponding tradeoffs* (Keeney and Raiffa, 1976), which does not require the value over money to be linear. To simplify the presentation, however, I maintain the stronger assumption.

(where $\theta_1, \theta_1 \in \Theta$), the WTP function can be used to choose among priced outcomes.

Naturally, elicitation of a WTP function is most intuitive when using direct monetary values. In other words, we elicit a function in which $L(p) = p$, so $v(A, p) = \bar{u}(A) + p$. We define a reference outcome (θ^0, p^0) , and assuming continuity of p , for any assignment $\hat{\theta}$ there exists a \hat{p} such that $(\hat{\theta}, \hat{p}) \sim (\theta^0, p^0)$.⁴ As $v(\cdot)$ is normalized such that $v(\theta^0, p^0) = 0$, \hat{p} is interpreted as the WTP for $\hat{\theta}$, or the *reserve price* of $\hat{\theta}$.

Theorem 3.7. *The WTP function is an MVF over differences in the reserve prices.*

Note that the WTP function is used extensively in economics, and that all the development in Section 3.2 could be performed directly in terms of WTP, relying on quasi-linearity for preference measurement, and without formalization using MVFs. This formalization however aligns this work with the fundamental difference independence theory by Dyer and Sarin.

In addition to facilitating the detection of GAI structure, the CDI condition supports elicitation using local queries, similar to how CAI is used by Braziunas and Boutilier (2005). I adopt their definition of *conditional set* of I_r , noted here S_r , as the set of neighbors of attributes in I_r not including the attributes of I_r . Clearly, S_r is the separating set of I_r in the CDI map, hence $\text{CDI}(I_r, V_r)$, where $V_r = A \setminus (I_r \cup S_r)$. From the definition of CDI, for any $I_r^1, I_r^2, V_r^1, V_r^2$ we have:

$$\bar{u}(I_r^2 S_r^0 V_r^1) - \bar{u}(I_r^1 S_r^0 V_r^1) = \bar{u}(I_r^2 S_r^0 V_r^2) - \bar{u}(I_r^1 S_r^0 V_r^2).$$

Eliciting the WTP function therefore amounts to eliciting the utility (WTP) of one full outcome (the reference outcome θ^0), and then obtaining the function over each maximal clique using monetary differences between its possible assignments, technique known as *pricing out* (Keeney and Raiffa, 1976; Clemen, 1996), keeping the variables in the conditional set fixed. These *ceteris paribus* elicitation queries are local in the sense that the decision maker does not need to consider the values of the rest of the attributes. Furthermore, in eliciting MVFs we can avoid the global scaling step that is required for vNM functions. Since the preference differences are extracted with respect to specific amounts of the attribute p , the utility is already scaled according to that external measure. Hence, once the conditional utility functions $\bar{u}([I_j])$ are obtained, we can calculate $\bar{u}(A)$ according to (3.5).

This last step may require (in the worst case) computation of a number of terms that is exponential in the number of max cliques. In practice however GAI decomposition is

⁴The term “reference outcome” is used here as a baseline for valuation of other outcomes, which is a different context than the “reference value” used for conditional utility.

expected to be sufficiently structured to limit the number of these terms (Braziunas and Boutilier, 2005). To take advantage of that we can use the search algorithm suggested by Braziunas and Boutilier (2005), which efficiently finds all the nonempty intersections for each clique.

In the rest of the work, when it is clear from the context that the utility function is an MVF scaled to express willingness-to-pay, as is the case in Chapters 4 and 5, I drop the bar from the notation of $\bar{u}(\cdot)$.

3.3.2 Optimization

As shown, the WTP function can be used directly for pairwise comparisons of priced outcomes. Another preference query often treated in the literature is *optimization*, or choice of best outcome, possibly under constraints.

Under FOPI, the unconstrained optimization of unpriced outcomes is trivial, because it amounts to a serial selection of the most desired value of each attribute. Therefore, I consider choice among attribute points with prices. Since any outcome can be best given enough monetary compensation, this problem is not well-defined unless the combinations are constrained somehow.

A particularly interesting optimization problem is the MMP (Definition 2.18), which arises in the context of auctions or negotiations between a buyer with utility $u_b(\cdot)$ and a seller with utility $u_s(\cdot)$. If $u_b(\cdot)$ and $u_s(\cdot)$ are each in a GAI form, then $u_b(A) - u_s(A)$ has the GAI form implied by the combined CDI map: one that includes the edges of the CDI map of $u_b(\cdot)$ as well as the edges of the CDI map of $u_s(\cdot)$. We can therefore employ well-studied combinatorial optimization procedures, such as variable elimination (e.g. Dechter (1996)), to find the best trading point under GAI structure (see also Gonzales and Perny (2005)). Similarly, this optimization can be done to maximize surplus between a trader's utility function and a pricing system that assigns a price to each level of each GAI element, and this way guide traders to their optimal bidding points. In Chapter 4 I develop a multiattribute procurement auction that builds on this idea.

3.3.3 An Example Scenario

My example purchase decision is of a vacation package, based on the TAC travel game (Wellman et al., 2007). In TAC (trading agent competition), clients provide their (additive) preference information to trading agents. The agents bid in a combination of auctions to purchase flight tickets, nights at hotels, and entertainment tickets. Their performance

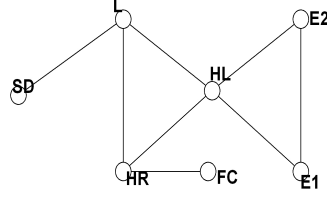


Figure 3.1: Map of CDI conditions. Utility function decomposes over maximal cliques.

is measured by the total surplus they provide to their clients. In a hypothetical multiattribute version of this scenario clients would provide agents with a compact representation of their full WTP function, and agents' performance would be measured with respect to that function.

In the original scenario, travel packages cover a given five-day period. My extended version includes the following attributes: *SD* (start day of trip, 1-3), *L* (length of stay, 2-4 nights), *HR* (hotel rating, 2-4 stars), *HL* (hotel location, 0-10 blocks away from the center), *E1* and *E2* (each can be one of two types of entertainment tickets, or none), *FC* (flight class, business or coach).

Assume the following reasonable set of difference independencies:

$$\text{CDI}(SD, \{HL, HR, E1, E2, FC\})$$

$$\text{CDI}(L, \{E1, E2, FC\})$$

$$\text{CDI}(HR, \{E1, E2\})$$

$$\text{CDI}(FC, \{HL, E1, E2\})$$

For example, the WTP and even preference order over the start date (*SD*) may depend on the length, but can be independent of the other attributes. Now looking at *HR*, we may be willing to pay more to relax in a fancier hotel after traveling coach, or if we are located far from the center (and therefore likely to spend more time at the hotel), or if we stay longer. However, this WTP does not depend on what type of entertainment we go to. Note that unconditional difference independence cannot be applied at all for this problem.

The CDI conditions are illustrated in Figure 3.1. Each pair of nodes that are CDI are not connected by an edge. The maximal cliques of this graph are $I_1 = \{SD, L\}$, $I_2 = \{L, HR, HL\}$, $I_3 = \{E1, E2, HL\}$, and $I_4 = \{FC, HR\}$.

The deepest nonempty intersections of cliques are pairwise. The local functions are

defined according to (3.5) as follows:

$$\begin{aligned}
u_1(I_1) &= u([SD, L]) \\
u_2(I_2) &= u([L, HR, HL]) - u([L]) \\
u_3(I_3) &= u([E1, E2, HL]) - u([HL]) \\
u_4(I_4) &= u([FC, HR]) - u([HR])
\end{aligned}$$

The number of entries required to represent $u(\cdot)$ under this decomposition is the sum of entries required for the functions $u([SD, L])$, $u([L, HR, HL])$, $u([E1, E2, HL])$, and $u([FC, HR])$ (note that the intersection domains are subsets of these domains), which is 213, whereas the full representation requires 5,346 entries, calculated as the full product of the sizes of the attributes' domains.

The function $u(\cdot)$ can be elicited, by the agents and from the clients, as follows. Initially the client is asked for her WTP for a reference package—for example $SD = 1$, $L = 2$, $HR = 2$, $HL = 10$, $E1 = \text{none}$, $E2 = \text{none}$, $FC = \text{coach}$. Then $u([FC, HR])$ is extracted: The client specifies her WTP to improve from a 2-star hotel to a 3-star one given that she is flying in coach. Then she provides her WTP to improve from coach to business class given 2 stars, and from 2 stars to 3 stars given business. Finally she specifies the WTP to improve from 3 stars to 4 stars given $FC = \text{coach}$ and then given $FC = \text{business}$. Since $S_4 = \{L, HL\}$, in all these queries the values of L and HL must be fixed on the reference points, and the values of $E1, E2, SD$ can be ignored. The number of queries for full elicitation is exponential in the size of the clique. Though it may still be too many for large problems, it is a better starting point for the application of heuristic techniques such as curving (eliciting a few sample points and fitting a function).

3.3.4 Beyond Quasi-Linearity

The quasi-linearity assumption in the money attribute is a key in the modeling of willingness-to-pay above. It in fact includes two assumptions: first, that the utilities of money and the attributes are additively separable, and second that the utility of money is linear. Linearity of money may be violated when large amounts of money are involved (due to diminishing returns), and I therefore relax this assumption in this section. The assumption that I still maintain is formalized as follows.

Definition 3.4. *Let \preceq be a preference relation over $A \cup \{p\}$. Then \preceq is additive in p if \preceq*

can be represented by

$$v(A, p) = u(A) + v_p(p).$$

Keeney and Raiffa (1976) describe a necessary and sufficient condition, called the *corresponding tradeoffs condition*, for such binary additive representation to hold. In essence, it holds if A and p can be transformed to scales ($u(A)$ and $v_p(p)$) on which their marginal rate of substitution is constant. Keeney and Raiffa suggest verification techniques for this condition.

Now assuming only that preferences are additive in p , $u(A)$ is still the WTP function, and

$$(A^1, p') \preceq (A^2, p'') \Leftrightarrow u(A^1) - u(A^2) \leq v_p(p'') - v_p(p').$$

Definition 3.5. Let A^1, A^2, A^3 , and A^4 be points in the domain of A . Then

$$[A^1, A^2] \succeq [A^3, A^4]$$

if and only if for any values p^1, p^2, p^3, p^4 of p such that $(A^1, p^1) \sim (A^2, p^2)$ and $(A^3, p^3) \sim (A^4, p^4)$,

$$v_p(p^1) - v_p(p^2) \leq v_p(p^3) - v_p(p^4).$$

Corollary 3.8. $u(A)$ is an MVF representing \succeq .

As in the quasi-linear case, we can obtain the reserve price p^i such that $v(A^i, p^i) = 0$.

To illustrate how this WTP function supports nonlinearity, assume that Alice is looking to buy a new car, and has the following reserve prices.

- Subaru with some configuration \hat{S} : \$20000.
- Subaru with \hat{S}_{pl} (\hat{S} with power locks): \$20100.
- Mercedes with configuration \hat{M} : \$40000.
- Mercedes with configuration \hat{M}_g (\hat{M} with GPS): \$40090.

If Alice's preferences over money exhibit diminishing evaluation, it may be the case that $v_p(w - 20100) - v_p(w - 20000) < v_p(w - 40090) - v_p(w - 40000)$ (w represents Alice's current wealth). In that case $[\hat{S}_{pl}, \hat{S}] \preceq^* [\hat{M}_g, \hat{M}]$, even though the monetary difference implies the opposite.

Now if \hat{M}_g costs 35100 and \hat{S} costs 15000, which of the two options should Alice choose? she would choose \hat{M}_g if and only if

$$u(\hat{M}_g) - u(\hat{S}) = v_p(w - 40090) - v_p(w - 20000) \geq v_p(w - 35100) - v_p(w - 15,000).$$

Again this formalization of the WTP as an MVF lets us use the compact representations developed in Section 3.2 to represent $u(A)$. Representation of $v_p(p)$ is less of a concern since it is a single-dimensional function, usually with a well defined curve.

Elicitation using pricing-out can still be used, however taking non-linearity into account. We start from the reference outcome (A^0, p^0) , $v(A^0, p^0) = 0$, and normalize $u(A)$ such that $u(A^0) = 0$ by compensating through $v_p(p)$. That also ensures $v_p(p^0) = 0$. Elicitation of each $u([I_j])$ is still aimed at finding the reserve price of each outcome, using the same local queries as those in the quasi-linear case: WTP differences over the assignments to the cliques I_j , keeping S_j fixed, and ignoring the value of V_j . However here the result is in terms of $v_p(p)$, rather than p . Assuming we obtained $v_p(p)$, $u(A)$ can be calculated again using (3.5).

This shown that with a relatively reasonable effort we can avoid the assumption that preferences over money are linear. However, to simplify the analysis of mechanisms proposed in this work I retain the quasi-linearity assumption throughout.

Chapter 4

Structured Multiattribute Auctions

4.1 Introduction

In this chapter I propose a multiattribute auction that addresses problems discussed in Section 2.3. In particular, no previous welfare-maximizing mechanism is at the same time *expressive* (i.e., allows traders to express non-additive preferences), *tractable* (does not depend on the fully exponential domain), and preserves buyer's *privacy of information*. In order to propose a mechanism that exhibits these properties, I combine previous approaches to multiattribute procurement with preference representation tools introduced in Chapter 3. I suggest a multiattribute mechanism that decreases elicitation and computation burden, approximately maximizes social welfare of the outcome, and creates an open competition between suppliers over a multi-dimensional domain. This mechanism is inspired by the auction design approach of Parkes and Kalagnanam (2005) (PK), that employs a price structure to drive traders to their efficient configurations. This allows us to avoid the sometimes unrealistic burden on the buyer to reveal his full utility function to sellers prior to bidding (see Section 2.3), which is required by most previous multiattribute auction literature. PK suggest two mechanisms, discussed in details in Section 2.3, which represent two extremes: NLD makes no assumption on traders' preference structure, whereas AD assumes fully additive preferences.

I motivate the approach I take in comparison to PK using the following example. A procurement department considers purchasing new hard drives (HD) for the desktops of a large number of employees, for example to allow more storage space, better performance, and to take advantage of a quantity discount. The company cares about several characteristics of the hard drives and the contract, listed below. Each attribute is listed with a designated attribute name (the first letter), and its domain. In some cases (e.g., attribute C) I use arbitrary number or letters to represent domain values. We can assume that these values have some meaningful interpretation.

RPM (R) 3600, 4200, 5400 rpm

Transfer rate (T) 3.4, 4.3, 5.7 MBS

Volume (V) 60, 80, 120, 160 GB

Compatibility (C) (with various types of desktops) 1, 2, 3

Quality rating (Q) (of the HD brand) 1, 2, 3, 4, 5

Delivery time (D) 10, 15, 20, 25, 30, 35 days

Warranty (W) 1, 2, 3 years

Insurance (I) (for the case the deal is signed but not implemented) α, β, γ

Payment timeline (P) 10, 30, 90 days

In practice, application scenarios may be extremely more complex, in both the number of attributes and the domains of each attribute (see the KLM example in Section 2.4). But even the simplified domain above poses a serious challenge for an auction such as NLD: the full domain has 87,480 distinct configurations, and the auction must obtain the buyer's valuation for each, maintain a price for each, and solicit bids for each. On the other hand, I argue below that using an additive auction such as AD misses important interdependencies among the attributes.

Consider, for example, the pair of attributes RPM and Volume. If the RPM is low (e.g., 3600), it is not desirable to have a very large hard drive such as 160GB, because the performance is going to be particularly poor. Even if some people would always prefer the larger HD, it is unreasonable to assume that their *willingness-to-pay* for the extra gigabytes does not depend on the RPM. A similar kind of relationship will typically exist between Volume and Transfer rate. Note that these pairs of attribute are complements (Definition 3.1).

Furthermore, I find it reasonable that Volume complements both Quality and Warranty. Larger hard drives are more prone to failures, therefore a larger one is more valuable when the quality is known to be high or if a longer warranty is promised. The use of the word "or" in the previous sentence implies another type of relationship: Quality and Warranty are very likely to be substitutes. Other reasonable dependencies are between Delivery, Insurance and Payment Terms (e.g., later delivery requires better insurance, later payment reduces the need of insurance) and perhaps between Insurance and Warranty (if the insurance covers the case that the supplier runs out of business during the warranty period). The corresponding CDI map is depicted in Figure 4.1a. As described in Chapter 3, the utility

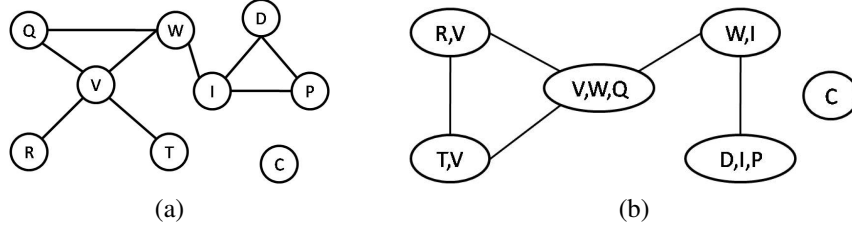


Figure 4.1: HD procurement problem: (a) CDI map, (b) GAI network.

function decomposes over the maximal cliques of its CDI map. It is therefore useful to view the clique graph of the map, or the GAI network (Gonzales and Perny, 2004). The corresponding GAI network is shown in Figure 4.1b.

This is by no means the exact set of dependencies that holds for the preferences of any procurement department faced with this problem. For example, RPM and Transfer Rate may also have a complementarity relationship between them, because one can be utilized better when the value of the other is higher. In this example, we might say that this relationship is weak enough to allow us to ignore it.

To illustrate the need to support GAI preferences I consider the case of traders with non-additive preferences bidding in an additive price space such as in auction AD (Parkes and Kalagnanam, 2005). If the buyer's preferences are not additive, choosing preferred levels per attribute (as in auction AD) admits undesired combinations and fails to guide the sellers to the efficient configurations. Non-additive sellers face an *exposure problem*, somewhat analogous to traders with complementary preferences that participate in simultaneous auctions (Wellman et al., 2008).¹ A value a^1 for attribute a may be optimal given that the value of another attribute b is b^1 , and arbitrarily suboptimal given other values of b . Therefore bidding a^1 and b^1 may result in a poor allocation if the seller is outbid on b^1 but left winning a^1 .

Instead of assuming full additivity, the auction designer can come up with a GAI preference structure that captures the set of common interdependencies among attributes. If traders could bid on clusters of interdependent attributes, it would solve the problems above. For example, if a and b are interdependent (meaning $\text{CDI}(a, b)$ does not hold), we should be able to bid on the cluster ab . If b in turn depends on c , we need another cluster bc . This is still better than a general pricing structure that solicits bids for the cluster abc . I stress that each trader may have a different set of interdependencies, and therefore to be

¹If only the sellers are non-additive, the auction design could potentially alleviate this problem by collecting a new set of bids each round, and also guiding non-additive sellers to bid on only one level per attribute in order to avoid undesired combinations.

completely general the GAI structure needs to account for all.² However, in practice many domains have natural dependencies that are mutual to traders.

4.2 GAI Auctions

In the procurement setting a single buyer wishes to procure a single good, in some configuration $\theta \in \Theta$ from one of the candidate sellers s_1, \dots, s_m . The buyer has some private valuation function (WTP) $u_b : \Theta \rightarrow \mathbb{R}^+$, and similarly each seller s_i has a private valuation function (willingness-to-accept), or cost function, c_i . The multiattribute allocation problem (Parkes and Kalagnanam, 2005) is the welfare optimization problem in procurement over a discrete domain (Definition 2.17), and the multiattribute matching problem (MMP) is a subproblem of MAP. $MMP(b, s_i)$ is defined to be a configuration that maximizes the surplus between the reported valuations of the buyer b and a seller s_i .

4.2.1 GAI Preferences in Procurement

Assume that preferences of all traders are reflected in a GAI structure I_1, \dots, I_g . I call each I_r a GAI *element*, and any assignment to I_r a *sub-configuration*. I use θ_r to denote the sub-configuration formed by projecting configuration θ to element I_r .

Definition 4.1. Let α be an assignment to I_r and β an assignment to $I_{r'}$. The sub-configurations α and β are consistent if for any attribute $a_j \in I_r \cap I_{r'}$, α and β agree on the value of a_j . A collection Φ of sub-configurations is consistent if all pairs $\alpha, \beta \in \Phi$ are consistent. The collection is called a cover if it contains exactly one sub-configuration α_r corresponding to each element I_r , $r \in \{1, \dots, g\}$.

Note that a consistent cover $\{\alpha_1, \dots, \alpha_g\}$ represents a full configuration, which I denote by $(\alpha_1, \dots, \alpha_g)$.

For example, consider the procurement of a good with three attributes, a, b, c . Each attribute's domain has two values (e.g., $\{a^1, a^2\}$ is the domain of a). Let the GAI structure be $I_1 = \{a, b\}, I_2 = \{b, c\}$. Figure 4.2 shows the simple CDI map and the corresponding GAI network, which is a GAI tree. Here, sub-configurations are assignments of the form $a^1 b^1$, $a^1 b^2$, $b^1 c^1$, and so on. The set of sub-configurations $\{a^1 b^1, b^1 c^1\}$ is a consistent cover, corresponding to the configuration $a^1 b^1 c^1$. In contrast, the set $\{a^1 b^1, b^2 c^1\}$ is inconsistent.

²This requirement is relaxed in Section 4.3.3.

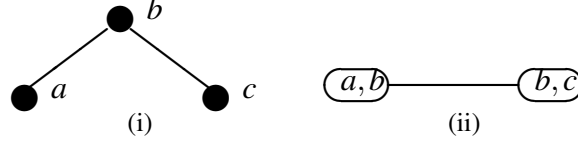


Figure 4.2: An example CDI map and GAI network. (i) CDI map for $\{a, b, c\}$, reflecting the single condition $\text{CDI}(a, c)$. (ii) The corresponding GAI network.

In this work I propose auctions with *GAI pricing structure*, that is a structure that assigns a price for each sub-configuration in a given GAI decomposition of Θ . A crucial property of this pricing scheme is that if the GAI price structure matches the trader's GAI structure, then traders do not face an exposure problem: no suboptimal configurations can be constructed from a set of optimal configurations. More precisely, if Ψ is a set of profit-maximizing configurations for a trader, and Φ is the collection of sub-configurations in Ψ , then any consistent cover over Φ is optimal as well.

This property can be ensured (see Lemmas 4.1 and 4.3) only if the GAI network is in the form of a tree or a forest (the *GAI tree*). A tree structure can be achieved for any set of CDI conditions by triangulation of the CDI map prior to construction of the clique graph (GAI networks and GAI trees are defined by Gonzales and Perny (2004), who also describe the triangulation algorithm). In the HD example above, we can ensure the tree structure by adding a dependency between RPM and Transfer rate, which would lead to a merge of the nodes (R, V) and (T, V) into (R, T, V) in the GAI network.

Under GAI, the buyer's value function u_b and sellers' cost functions c_i can be decomposed as in (3.4). I use $f_{b,r}$ and $f_{i,r}$ to denote the local functions of buyer and sellers (respectively), according to (3.5).

4.2.2 The GAI Auction

I define an iterative multiattribute auction that maintains a GAI pricing structure: that is, in any round t , there is a price $p^t(\cdot)$, corresponding to each sub-configuration of each GAI-tree element. The price $p^t(\theta)$ of a configuration θ at round t is defined in terms of the sub-configuration prices and a global discount term Δ ,

$$p^t(\theta) = \sum_{r=1}^g p^t(\theta_r) - \Delta.$$

Bidders submit *sub-bids* on sub-configurations and on the global discount Δ .³ Sub-bids

³The discount term could be replaced with a uniform price reduction across all sub-configurations.

are submitted in each round and they expire in the next round. A sub-bid in round t for configuration θ_r is automatically assigned the price $p^t(\theta_r)$. The set of *full bids* of a seller contains all consistent covers that can be generated from that seller's current set of sub-bids. The existence of a full bid over a configuration θ represents the seller's willingness to accept the price $p^t(\theta)$ for supplying θ .

At the start of the auction, the buyer reports (to the auction, not to sellers) a complete valuation function $u_b(\cdot)$ in GAI form $(f_{b,1}, \dots, f_{b,g})$, such that $u_b(\theta) = \sum_r f_{b,r}(\theta_r)$. The initial prices of sub-configurations are set at some level above the buyer's valuations, that is, $p^1(\theta_r) > f_{b,r}(\theta_r)$ for all θ_r . The discount Δ is initialized to zero. The auction has the dynamics of a *descending clock auction*: at each round t , bids are collected for current prices and then prices are reduced according to price rules. A seller is considered *active* in a round if she submits at least one full bid. In round $t > 1$, only sellers who were active in round $t - 1$ are allowed to participate, and the auction terminates when no more than a single seller is active. I denote the set of sub-bids submitted by s_i by \mathcal{B}_i^t , and the corresponding set of full bids is

$$B_i^t = \{\theta = (\theta_1, \dots, \theta_g) \in \Theta \mid \{\theta_1, \dots, \theta_g\} \subseteq \mathcal{B}_i^t\}.$$

In the example, a seller could submit sub-bids on a set of sub-configurations such as $\{a^1 b^1, b^1 c^1\}$, and that combines to a full bid on $a^1 b^1 c^1$.

The auction proceeds in two phases. In the first phase (A), at each round t the auction computes a set of *buyer-preferred* sub-configurations \mathcal{M}^t . Section 4.3.1 shows how to define \mathcal{M}^t to ensure convergence, and Section 4.3.2 shows how to efficiently compute it.

In phase A, the auction adjusts prices after each round, reducing the price of every sub-configuration that has received a bid but is not in the preferred set. Let ε be the prespecified price increment parameter. Specifically, the phase A price change rule is applied to all $\theta_r \in \bigcup_{i=1}^n B_i^t \setminus \mathcal{M}^t$:

$$p^{t+1}(\theta_r) \leftarrow p^t(\theta_r) - \frac{\varepsilon}{g}. \quad [\text{A}]$$

Let M^t denote the set of configurations that are consistent covers in \mathcal{M}^t :

$$M^t = \{\theta = (\theta_1, \dots, \theta_g) \in \Theta \mid \{\theta_1, \dots, \theta_g\} \subseteq \mathcal{M}^t\}.$$

The auction switches to phase B when all active sellers have at least one full bid in the buyer's preferred set:

$$\forall i. B_i^t = \emptyset \vee B_i^t \cap M^t \neq \emptyset. \quad [\text{SWITCH}]$$

Let T be the round at which [SWITCH] becomes true. At this point, the auction selects the

buyer-optimal full bid η_i for each seller s_i .

$$\eta_i = \arg \max_{\theta \in B_i^T} (u_b(\theta) - p^T(\theta)). \quad (4.1)$$

In phase B, s_i may bid only on η_i . The prices of sub-configurations are fixed at $p^T(\cdot)$ during this phase. The only adjustment in phase B is to Δ , which is increased in every round by ε . The auction terminates when at most one seller (if exactly one, designate it $s_{\hat{i}}$) is active. There are four distinct cases:

1. All sellers drop out in phase A (i.e., before rule [SWITCH] holds). The auction terminates with no allocation.
2. All active sellers drop out in the same round in phase B. The auction selects the best seller from the preceding round as $s_{\hat{i}}$, and applies the applicable case below.
3. The auction terminates in phase B with a final price above the buyer's valuation, $p^T(\eta_{\hat{i}}) - \Delta > u_b(\eta_{\hat{i}})$. The auction offers the winner $s_{\hat{i}}$ an opportunity to supply $\eta_{\hat{i}}$ at price $u_b(\eta_{\hat{i}})$.
4. The auction terminates in phase B with a final price $p^T(\eta_{\hat{i}}) - \Delta \leq u_b(\eta_{\hat{i}})$. This is the ideal situation, where the auction allocates the chosen configuration and seller at this resulting price.

The overall auction is described by high-level pseudocode in Procedure GAI-Auction. The role of phase A is to guide the traders to their efficient configurations (MMP solutions): prices are reduced on configurations that are *optimal for a seller* but not preferred by the buyer, making it either less attractive to the seller or preferred by the buyer. Phase B is a one-dimensional competition over the surplus that remaining seller candidates can provide to the buyer. In the next section I formalize the notions mentioned above: optimal for the seller, and buyer-preferred, prove that phase A indeed converges, and that phase B selects a seller whose efficient configuration yields (approximately) the highest surplus. In Section 4.3.2 I discuss the computational tasks associated with the auction, and Section 4.4 provides a detailed example.

4.3 Analysis

4.3.1 Economic Properties

When the optimal solution to *MAP* (2.7) provides negative welfare and sellers do not bid below their cost, the auction terminates in phase A, no trade occurs and the auction is trivially efficient. I therefore assume throughout the analysis that the optimal (seller, configuration) pair provides non-negative welfare. Furthermore, efficiency results below hold

Procedure:GAI Auction

Collect a reported valuation, $u_b(\cdot)$ from the buyer;
 Set high initial prices, $p^1(\theta_r)$ on each sub-configuration θ_r , and set $\Delta = 0$;
while *not [SWITCH]* **do**
 Collect sub-bids from sellers;
 Compute \mathcal{M}^t ;
 Apply price change by [A];
end
 Compute η_i ;
while *more than one active seller* **do**
 Increase Δ by ε ;
 Collect bids on (η_i, Δ) from sellers;
end
 Implement allocation and payment to winning seller;

when the buyer is truthful (as reasoned in Section 2.3), meaning that $u_b(\cdot)$ is her real utility function. Another way to interpret the efficiency results is that they apply to the face value of the buyer's report. At this point I do not assume truthful sellers.

The buyer profit from a configuration θ is defined as⁴

$$\pi_b(\theta) = u_b(\theta) - p(\theta),$$

and similarly $\pi_i(\theta) = p(\theta) - c_i(\theta)$ is the profit of seller s_i . In addition, for $\mu \subseteq \{1, \dots, g\}$, I denote the corresponding set of sub-configurations by θ_μ , and define the profit from a configuration θ over the subset μ as

$$\pi_b(\theta_\mu) = \sum_{r \in \mu} (f_{b,r}(\theta_r) - p(\theta_r)).$$

$\pi_i(\theta_\mu)$ is defined similarly for s_i . Crucially, for any μ and its complement $\bar{\mu}$ and for any trader τ ,

$$\pi_\tau(\theta) = \pi_\tau(\theta_\mu) + \pi_\tau(\theta_{\bar{\mu}}).$$

The function $\sigma_i : \Theta \rightarrow \Re$ represents the welfare, or surplus function $u_b(\cdot) - c_i(\cdot)$. For any price system p ,

$$\sigma_i(\theta) = \pi_b(\theta) + \pi_i(\theta).$$

Definition 4.2. *A seller is called a straightforward bidder (SB) if at each round t she bids*

⁴I drop the t superscript in generic statements involving price and profit functions, understanding that all usage is with respect to the (currently) applicable prices.

on \mathcal{B}_i^t as follows: if $\max_{\theta \in \Theta} \pi_i^t(\theta) < 0$, then $\mathcal{B}_i^t = \emptyset$. Otherwise let

$$\begin{aligned} \emptyset \subset \Omega_i^t &\subseteq \arg \max_{\theta \in \Theta} \pi_i^t(\theta) \\ \mathcal{B}_i^t &= \{\theta_r \mid \theta \in \Omega_i^t, r \in \{1, \dots, g\}\}. \end{aligned}$$

Intuitively, an SB seller follows a *myopic best response* strategy (MBR), meaning they bid myopically rather than strategically by optimizing their profit with respect to current prices. Note that SB sellers may bid on any nonempty *subset* of their myopically optimal configurations.

In order to calculate \mathcal{B}_i^t , sellers need to optimize their current profit function, as discussed in Section 3.3.2.

The following lemma shows that sellers do not face the exposure problem described earlier.

Lemma 4.1. *Let Ψ be a set of configurations, all maximizing profit for a trader τ (seller or buyer) at the relevant prices. Let $\Phi = \{\theta_r \mid \theta \in \Psi, r \in \{1, \dots, g\}\}$. Then any consistent cover in Φ is also a profit-maximizing configuration for τ .*

The proof (in Appendix A) relies on the tree structure of the GAI network.

Corollary 4.2. *For SB seller s_i ,*

$$\forall t, \forall \theta' \in \mathcal{B}_i^t, \pi_i^t(\theta') = \max_{\theta \in \Theta} \pi_i^t(\theta).$$

Next I consider combinations of configurations that are only within some δ of optimality.

Lemma 4.3. *Let Ψ be a set of configurations, all are within δ of maximizing profit for a trader τ at the prices, and Φ defined as in Lemma 4.1. Then any consistent cover in Φ is within δg of maximizing utility for τ .*

This bound is tight, that is for any GAI tree and a non-trivial domain we can construct an example set Ψ as above in which there exists a consistent cover whose utility is exactly δg below the maximal.

Next I formally define \mathcal{M}^t . For connected GAI trees, \mathcal{M}^t is the set of sub-configurations that are part of a configuration within ε of optimal. When the GAI tree is in fact a forest, we apportion the error proportionally across the disconnected trees. Let G be comprised of trees G_1, \dots, G_h . I use θ_j to denote the projection of a configuration θ on the tree G_j , and g_j denotes the number of GAI elements in G_j .

$$\mathcal{M}_j^t = \{\theta_r \mid \pi_b^t(\theta_j) \geq \max_{\theta'_j \in \Theta_j} \pi_b^t(\theta'_j) - g_j \frac{\varepsilon}{g}, r \in G_j\}$$

Then define $\mathcal{M}^t = \bigcup_{j=1}^h \mathcal{M}_j^t$.

Let $e_j = g_j - 1$ denote the number of edges in G_j . I define the connectivity parameter, $e = \max_{j=1, \dots, h} e_j$. As shown below, this connectivity parameter is an important factor in the performance of the auction.

Corollary 4.4.

$$\forall \theta' \in M^t, \pi_b^t(\theta') \geq \max_{\theta \in \Theta} \pi_b^t(\theta) - (e+1)\varepsilon$$

Lemmas 4.5 through 4.8 show that through the price system, the choice of buyer preferred configurations, and price change rules, phase A leads the buyer and each of the sellers to their mutually efficient configuration.

Lemma 4.5. $\max_{\theta \in \Theta} \pi_b^t(\theta)$ does not change in any round t of phase A.

The proof shows that if the buyer profit of a configuration θ increases, due to price reduction, then the price change rules ensure that the profit of θ must have been smaller than the maximal profit, by at least the amount of the price reduction.

Lemma 4.6. The price of at least one sub-configuration must be reduced at every round in phase A.

The proof connects the price reduction rule with the choice of buyer-preferred set: in phase A there must be a seller that bids on a configuration that has an element that is not in the preferred set, and therefore its price must be reduced.

Lemma 4.7. When the solution to MAP provides positive surplus, and at least the best seller is SB, the auction must reach phase B.

This leads to the following key result.

Lemma 4.8. For SB seller s_i , η_i is $(e+1)\varepsilon$ -efficient, that is its surplus is within $(e+1)\varepsilon$ of the surplus provided by $\text{MMP}(b, s_i)$.

In the fully additive case this loss of efficiency reduces to ε . On the other extreme, if the GAI network is connected then $e+1 = g$. Also note that if we do not assume any preference structure, meaning that the CDI map is fully connected, $g = 1$ and the efficiency loss is again ε . The proof is straightforward: η_i maximizes the profit of the s_i , and at the same prices approximately maximizes the profit of the buyer. This establishes the approximate bilateral efficiency of the results of phase A (at this point under the assumption of SB).

Based on phase B's simple role as a single-dimensional bidding competition over the discount, I next assert that the overall result is efficient under SB, which in turn proves to be an approximately ex-post equilibrium strategy in the two phases.

Lemma 4.9. *If sellers s_i and $s_{\tilde{i}}$ are SB, and s_i is active at least as long as $s_{\tilde{i}}$ is active in phase B, then*

$$\sigma_i(\eta_i) \geq \max_{\theta \in \Theta} \sigma_{\tilde{i}}(\theta) - (e+2)\varepsilon.$$

Theorem 4.10. *Given a truthful buyer and SB sellers, the auction is $(e+2)\varepsilon$ -efficient: the surplus of the final allocation is within $(e+2)\varepsilon$ of the maximal surplus.*

Following PK, I rely on an equivalence to the one-sided VCG auction to establish incentive properties for the sellers. In the one-sided multiattribute VCG auction, the buyer reports valuation u_b , the sellers report costs functions \hat{c}_i , and the buyer pays the sell-side VCG payment to the winning seller.

Definition 4.3. *Let (θ^*, i^*) be an optimal solution to MAP. Let $(\tilde{\theta}, \tilde{i})$ be the best solution to MAP when i^* does not participate. The sell-side VCG payment is*

$$VCG(u_b, \hat{c}_i) = u_b(\theta^*) - (u_b(\tilde{\theta}) - \hat{c}_{\tilde{i}}(\tilde{\theta})).$$

It is well-known that truthful bidding is a dominant strategy for sellers in the one-sided VCG auction. It is also shown by PK that the maximal regret for buyers from bidding truthfully in this mechanism is $u_b(\theta^*) - c_{i^*}(\theta^*) - (u_b(\tilde{\theta}) - \hat{c}_{\tilde{i}}(\tilde{\theta}))$, that is, the *marginal product* of the efficient seller.

Usually in iterative auctions the VCG outcome is only nearly achieved, but the deviation is bounded by the minimal price change. I show a similar result, and therefore define δ -VCG payments.

Definition 4.4. *A Sell-side δ -VCG payment for MAP is a payment p such that*

$$VCG(u_b, \hat{c}_i) - \delta \leq p \leq VCG(u_b, \hat{c}_i) + \delta.$$

Lemma 4.11. *When sellers are SB, the payment at the end of GAI auction is sell-side $(e+2)\varepsilon$ -VCG.*

When payment is guaranteed to be δ -VCG sellers can affect their payment only within that range, therefore their gain by falsely reporting their cost is bounded by 2δ .

Theorem 4.12. *SB is $(3e+5)\varepsilon$ ex-post Nash equilibrium for sellers in the GAI auction. That is, sellers cannot gain more than $(3e+5)\varepsilon$ by deviating from SB, given that other sellers follow SB.*

However, in order to exploit this potential gain sellers need to know, for a given configuration in M^t , whether it was explicitly selected as approximately optimal for the buyer, or it is a combination of sub-configurations from approximately optimal configurations. It seems highly unlikely for sellers to have such information. They are more likely to lose if they do not bid on their myopically optimal configurations.

4.3.2 Computation and Complexity

The size of the price space maintained in the auction is equal to the total number of sub-configurations, meaning it is exponential in $\max_r |I_r|$. This is also equivalent to the *tree-width* (plus one) of the original CDI map. For the purpose of the computational analysis, I define $I = \bigcup_{r=1}^g \prod_{a_j \in I_r} D(a_j)$, the collection of all sub-configurations. The first purpose of this sub-section is to show that the complexity of all the computations required for the auction depends only on $|I|$, i.e., no computation depends on the size of the full exponential domain.

I first consider the computation of \mathcal{M}^t . Since M^t grows monotonically with t , a naive application of optimization algorithm to generate the best outcomes sequentially might end up enumerating significant portions of the fully exponential domain. However as shown below this plain enumeration can be avoided.

Proposition 4.13. *The computation of \mathcal{M}^t can be performed in time $O(|I|^2)$. Moreover, the total time spent on this task throughout the auction is $O(|I|(|I| + T))$.*

The bounds are in practice significantly lower, based on results on similar problems from the probabilistic reasoning literature (Nilsson, 1998).

One of the benefits of the compact pricing structure is the compact representation it lends for bids: sellers submit only sub-bids, and therefore the number of them submitted and stored per seller is bounded by $|I|$. Several additional computation tasks are performed by the auction: $B_i^t \neq \emptyset$, rule [SWITCH], and choice of η_i . All these tasks involve the set B_i^t , but their performance depends only on the size of the (potentially more compact) set \mathcal{B}_i^t , because they can all be conducted by performing combinatorial optimization over \mathcal{B}_i^t and $\mathcal{B}_i^t \cap \mathcal{M}^t$.

Next, I analyze the number of rounds it takes for the auction to terminate. Phase B requires $\max_{i \in \{1, \dots, n\}} \pi_i^T(\eta_i) \frac{1}{\epsilon}$. Since this is equivalent to price-only auctions, the concern is only with the time complexity of phase A. We can obtain a bound on the number of rounds of phase A by comparing the sum of prices of all sub-configurations in round 1 and in round T . By Lemma 4.5, and assuming no negative valuations, this sub of prices in

round T cannot be negative. Also, in each round at least one price is reduced by $\frac{g}{\epsilon}$. Hence an upper bound on the number of rounds required is

$$T \leq \sum_{\theta_r \in I} p^1(\theta_r) \frac{g}{\epsilon}.$$

This bound is rather loose, and its purpose is to ensure that the number of rounds does not depend on the size of the fully exponential domain. It depends on the number of sub-configurations, and on the result of dividing the initial price by the minimum price decrement.

Usually Phase A will converge much faster. Let the initial negative profit chosen by the auctioneer be $m = \max_{\theta \in \Theta} \pi_b^1(\theta)$. In the worst case, phase A needs to run until $\forall \theta \in \Theta. \pi_b(\theta) = m$. This happens for example when $\forall \theta_r \in I. p^1(\theta_r) = f_{b,r}(\theta_r) + \frac{m}{g}$. In general, the closer the initial prices reflect buyer valuation, the faster phase A converges. One extreme is to choose $p^1(\theta_r) = f_{b,r}(\theta_r) + \frac{m}{g}$. That would make phase A redundant, at the cost of full initial revelation of buyer's valuation as done in other mechanisms (Section 2.3). Between this option and the other extreme, which is $\forall \alpha, \hat{\alpha} \in I, p^1(\alpha) = p^1(\hat{\alpha})$, the auctioneer has a range of choices to determine the right tradeoff between convergence time and information revelation. In the example below, the choice of a lower initial price for the domain of I_1 provides some speedup by revealing a harmless amount of information—that I_2 is less important than I_1 . In the simulations described later in this chapter I also set constant initial prices within each GAI element.

Another potential concern is the communication cost associated with the descending auction style. The sellers need to send their bids over and over again at each round. A simple change can be made to avoid much of the redundant communication: the auction can retain sub-bids from previous rounds on sub-configurations whose price did not change. Since combinations of sub-bids from different rounds can yield suboptimal configurations, each sub-bid should be tagged with the number of the latest round in which it was submitted, and only consistent combinations *from the same round* are considered to be full bids. With this implementation sellers need not resubmit their bid until a price of at least one of its sub-configurations has changed.

4.3.3 Assumptions on Preferences

A key aspect in implementing GAI based auctions is the choice of the preference structure, that is, the elements $\{I_1, \dots, I_g\}$. In some domains the structure can be more or less robust over time and over different decision makers. When this is not the case, extracting reliable

	I_1				I_2			
	a^1b^1	a^2b^1	a^1b^2	a^2b^2	b^1c^1	b^2c^1	b^1c^2	b^2c^2
f_b	65	50	55	70	50	85	60	75
f_1	35	20	30	70	65	65	70	61
f_2	35	20	25	25	55	110	70	95

Table 4.1: GAI utility functions for the example domain. f_b represents the buyer's valuation, and f_1 and f_2 costs of the sellers s_1 and s_2 .

structure from sellers (in the form of CDI conditions) is a serious challenge. This could have implications on the applicability of the auction for such domains, but in fact it can be overcome. It turns out that we can run this auction without any assumptions on sellers' preference structure. The only place where this assumption is used in the analysis is for Lemma 4.1. If sellers whose preference structure does not agree with the one used by the auction are guided to submit only one full bid at each round, or a set of bids that does not yield undesired consistent combinations, all the properties of the auction still hold. The configuration η_i obtained for seller s_i in the end of phase A is still bilaterally efficient: At the same set of prices, it is optimal for s_i and approximately optimal for the buyer.

It is therefore essential only that the buyer's preference structure is modeled accurately. The optimization problem solved by sellers in each round is more complex, but still manageable because they can use the union of their GAI structure with the auction's structure, as explained in Section 3.3.2. Of course, capturing sellers' structures as well is still preferred since it can speed up the execution and let sellers take advantage of the compact bid representation.

In both cases the choice of clusters may significantly affect the complexity of the price structure and the runtime of the auction. It is sometimes better to ignore some weaker interdependencies in order to reduce dimensionality. The complexity of the structure also affects the efficiency of the auction through the value of e .

4.4 Example

I use the example settings introduced in Section 4.2.1. Recall that the GAI structure is $I_1 = \{a, b\}, I_2 = \{b, c\}$ (note that $e = 1$). Table 4.1 shows the GAI utilities for the buyer and the two sellers s_1, s_2 . The efficient allocation is $(s_1, a^1b^2c^1)$: the buyer's valuation is $55 + 85 = 140$, the valuation of s_1 is $30 + 65 = 95$, hence the surplus is 45. The maximal surplus of the second best seller, s_2 , is 25, achieved by $a^1b^1c^1$, $a^2b^1c^1$, and $a^2b^2c^2$. I set

	I_1				I_2			
t	a^1b^1	a^2b^1	a^1b^2	a^2b^2	b^1c^1	b^2c^1	b^1c^2	b^2c^2
1	75	75	75	75	90	90	90	90
		s_1, s_2		*	s_1, s_2	*		
2	75	71	75	75	86	90	90	90
		s_2	s_1	*	s_2	*		s_1
3	75	67	71	75	82	90	90	86
		s_1, s_2		*	s_2	*	s_1	*
4	75	63	71	75	78	90	86	86
		s_2	s_1	*	s_2	$*, s_1$		$*, s_1$
5	75	59	67	75	74	90	86	86
	s_2		$*, s_1$	*	s_2	$*, s_1$		$*, s_1$
6	71	59	67	75	70	90	86	86
		s_2	$*, s_1$	*		$*, s_1$	s_2	$*, s_1$
7	71	55	67	75	70	90	82	86
	s_2		$*, s_1$	*	s_2	$*, s_1$		$*, s_1$
8	67	55	67	75	66	90	82	86
	*	s_2	$*, s_1$	*	*	$*, s_1$	s_2	$*, s_1$
9	67	51	67	75	66	90	78	86
	$*, s_2$		$*, s_1$	*	$*, s_2$	$*, s_1$		$*, s_1$

Table 4.2: Auction progression in phase A. Sell bids and designation of \mathcal{M}^t (using *) are shown below the price of each sub-configuration.

all initial prices over I_1 to 75, all initial prices over I_2 to 90, and $\varepsilon = 8$, meaning that price reduction for sub-configurations is 4. The difference in the valuations of the two sellers is lower than the potential efficiency loss of Theorem 4.10 (here $(e + 2)\varepsilon = 24$). However, in this case it is guaranteed that s_1 wins, either with the efficient allocation, or with $a^1b^2c^2$ which provides a surplus of 39. The reason is that these are the only two configurations of s_1 which are within $(e + 1)\varepsilon = 16$ of the solution to $MMP(b, s_1)$ (45), therefore by Lemma 4.8 one of them must be chosen as η_1 . Both of these configurations provide more than ε surplus over s_2 's most efficient configuration (and this is sufficient in order to win in phase B).

Table 4.2 shows the progress of phase A. Initially all configuration have the same cost (165), so sellers bid on their lowest cost configuration which is $a^2b^1c^1$ for both (with profit 80 to s_1 and 90 to s_2), and that translates to sub-bids on a^2b^1 and b^1c^1 . \mathcal{M}^1 contains the sub-configurations a^2b^2 and b^2c^1 of the highest value configuration $a^2b^2c^1$. Price is therefore decreased on a^2b^1 and b^1c^1 . After the price change, s_1 has higher profit (74) on $a^1b^2c^2$ and she therefore bids on a^1b^2 and b^2c^2 . Now (round 2) their prices go down, reducing the profit on $a^1b^2c^2$ to 66 and therefore in round 3 s_1 prefers $a^2b^1c^2$ (profit 67). After the next price change the configurations $a^1b^2c^1$ and $a^1b^2c^2$ both become optimal (profit 66),

and the sub-bids a^1b^2 , b^2c^1 and b^2c^2 capture the two. These configurations stay optimal for another round (5), with profit 62. s_2 however sticks to $a^2b^1c^1$ during the first four rounds, switching to $a^1b^1c^1$ in round 5. At this point s_1 has a full bid (in fact two full bids: $a^1b^2c^2$ and $a^1b^2c^1$) in M^5 , and she no longer changes her bids because the price of her optimal configurations does not decrease. It takes four more rounds for s_2 and \mathcal{M}^t to converge ($\mathcal{M}^{10} \cap B_2^{10} = \{a^1b^1c^1\}$).

After round 9 the auction sets $\eta_1 = a^1b^2c^1$ (which yields more buyer profit than $a^1b^2c^2$) and $\eta_2 = a^1b^1c^1$. For the next round (10) $\Delta = 8$, increased by 8 for each subsequent round. Note that $p^9(a^1b^1c^1) = 133$, and $c_2(a^1b^1c^1) = 90$, therefore $\pi_2^T(\eta_2) = 43$. In round 15, $\Delta = 48$ meaning $p^{15}(a^1b^1c^1) = 85$ and that causes s_2 to drop out, setting the final allocation to $(s_1, a^1b^2c^1)$ and $p^{15}(a^1b^2c^1) = 157 - 48 = 109$. That leaves the buyer with a profit of 31 and s_1 with a profit of 14, less than ε below the VCG profit 20.

The welfare achieved in this case is optimal. To illustrate how some efficiency loss could occur consider the case that $c_1(b^2c^2) = 60$. In that case, in round 3 the configuration $a^1b^2c^2$ provides the same profit (67) as $a^2b^1c^2$, and s_1 bids on both. While $a^2b^1c^2$ is no longer optimal after the price change, $a^1b^2c^2$ remains optimal on subsequent rounds because $b^2c^2 \in \mathcal{M}^t$, and the price change of a^1b^2 affects both $a^1b^2c^2$ and the efficient configuration $a^1b^2c^1$. When phase A ends $B_1^{10} \cap M^{10} = \{a^1b^2c^2\}$ so the auction terminates with the slightly suboptimal configuration and surplus 40.

In the rest of this chapter I describe an analysis of GAI auctions via simulations. The analysis considers economic efficiency, computational issues, and information revelation properties.

4.5 Experimental Design

The main idea behind the GAI auctions is to improve efficiency over the auctions that assume the additive representation, when the preferences are in fact not additive. However, the theoretical efficiency guarantee of GAI auctions depends on e , the connectivity of the GAI network. This implies a potential conflict: more accurate modeling improves efficiency with respect to the true utility, but may cause loss of efficiency due to higher connectivity. Therefore, an obvious goal for an experimental analysis is to test whether the more accurate preference modeling is indeed more efficient; in particular, whether GAI auctions are more efficient than additive auctions, given that the preferences are not additive. I assume that the buyer's preferences have some GAI structure, and compare the performance of the GAI auctions that model this structure with the performance of an auc-

tion that is restricted to an additive representation. As the auction *additive approximation* (AP), I use an instance of GAI auction in which the pricing structure is additive. This auction is in fact very similar to auction AD (Parkes and Kalagnanam, 2005). To the best of my knowledge, AD is the only welfare maximizing multiattribute auction that has been proposed based on additive preferences, beside those that require full revelation of the buyer's utility (see Section 2.3).

4.5.1 GAI Random Utility

For experimental purposes, I generate random utility functions to represent the buyer's value function and the sellers' cost functions. In order to imitate realistic, GAI-structured utility functions, I employ results by Braziunas and Boutilier (2005), who establish that GAI utility can be represented using locally normalized functions, weighed by scaling constants, similarly to additive utility. Let $\bar{u}_r(I_r) = u_r([I_r])$ denote a *local* utility function over I_r . The function $\bar{u}_r(I_r)$ is distinct for each r , so we can normalize each to $[0, 1]$. Next, let $\bar{f}_r(I_r)$ be defined using formula (3.5), with $u([I_r])$ replaced with $\bar{u}_r(I_r)$. Braziunas and Boutilier show that there exist scaling constants $\lambda_r \in [0, 1]$ such that

$$\bar{u}(S) = \sum_{r=1}^g \lambda_r \bar{f}_r(I_r). \quad (4.2)$$

The implication for the generation of random GAI utility functions is that we can draw the functions $\bar{u}_r(\cdot)$ independently over $[0, 1]$, and perform a global scaling afterwards. Note that (3.5) is much simplified for GAI trees, because each element I_r intersects with at most one preceding element, its parent $I_{p(r)}$:

$$\bar{f}_r = \bar{u}_r(I_r) - \bar{u}_r([I_r \cap I_{p(r)}]). \quad (4.3)$$

I refer to the functions $\bar{u}_r([I_r])$ as *subutility functions*. Note that values of the form $\bar{u}_r([I_r \cap I_{p(r)}])$ are drawn only once and used in both $\bar{u}_r(I_r)$ and $\bar{u}_r(I_{p(r)})$. This representation lets us draw random GAI functions, for a given GAI tree structure, using the following steps:

1. Draw random subutility functions $\bar{u}_r(I_r), r = 1, \dots, g$ in the range $[0, 1]$.
2. Compute $\bar{f}_r(\cdot), r = 1, \dots, g$ using (4.3).
3. Draw random scaling constants and compute $\bar{u}(S)$ by (4.2).

The scaling constants represent the importance that the decision maker accords to the corresponding GAI element in the overall decision. This procedure results in utilities that are normalized in $[0, 1]$. To accommodate means and variances of different agents, I then scale

$\bar{u}(\cdot)$ to $u(\cdot)$ in the desired range.

4.5.2 Structured Subutility

A subutility function in the model above may represent any valuation over the subspace. In practice we may often find additional structure within each GAI element. I introduce two structures which I consider most typical and generally applicable, and I use them for the simulations, along with completely random local functions.

As I argue in Section 2.1.2, typical purchase and sale decisions exhibit FOPI, under which most or all single attributes have a natural ordering of quality. For example, in a hard drive procurement setting, the buyer always prefers more memory, higher RPM, longer warranty, and so on. To implement FOPI, I let the integer values of each attribute represent its quality. For example, if a belongs to some GAI element $I_r = \{a, b\}$, I make sure that $\bar{u}_r(a_i, b') \geq \bar{u}_r(a_j, b')$ for any $a_i > a_j$, $a_i, a_j \in D(a)$, and any $b' \in D(b)$. This must of course hold in any GAI element that includes a , and for any attribute a that is (first-order) preferential independent. I enforce the condition after all the values for that GAI element have been drawn, through a special-purpose sorting procedure, applied between steps 1 and 2 above.

The FOPI condition makes the random utility function more realistic, and in particular more appropriate to the target application. Once attributes exhibit FOPI, the dependencies among different attributes are likely to be framed as complements or substitutes (Definition 3.1). This relationship between attributes is ruled out under an additive utility function, but (as discussed in Chapter 3) admitted under mutual utility independence (MUI, Definition 2.11). MUI is a very demanding condition, and imposes far more structure than FOPI. It is however, still more flexible than the additive form.

A set S' of MUI attributes may still include complements or substitutes, but only if the same type of condition (complementarity or substitutivity), with the same strength, occurs for every two attributes in S' . Therefore, given MUI assumption, we can use a single variable to control the level of complementarity or substitutivity within each GAI element, as shown in Chapter 3. If a set S' is MUI, it can be represented using a single-dimensional subutility function, and $|S'| + 1$ constants. The first $|S'|$ constants, $k_1, \dots, k_{|S'|}$, (roughly) determine the weights of the attributes. The last constant, which is the MUI-factor k , is computable from $k_1, \dots, k_{|S'|}$ (Keeney and Raiffa, 1976). According to Theorem 3.2, k can be regarded as a measurement of substitutivity or complementarity: if $-1 < k < 0$, the attributes of S' are substitutes, if $k > 0$ they are complements, and if $k = 0$ they are additively independent. The more extreme the value of k is within the range (the further away it is

from zero), the stronger is the substitutivity or complementarity.

In an elicitation procedure, one would normally extract the $|S'|$ scaling constants from a user, and then compute k according to Definition 2.12 (Keeney and Raiffa, 1976). For my purposes, I first determine k according to the relationship I wish to impose on the attributes, and then draw MUI scaling constants that are consistent with this value. More explicitly, I draw random scaling constants, and then iteratively modify all the constants, until a set of constants is found that is consistent with k . The next step is to compute $\bar{u}_r(I_r)$ according to the MUI formula (Equation 2.3). The $\bar{u}_r(I_r)$ (for all r) are in the range $[0, 1]$, hence at this point we can proceed with steps 2 and 3 above. Note that in this procedure several distinct sets of scaling constants are used: the g constants used in step 3 scale the different GAI elements, whereas the MUI constants, per GAI element, scale the attributes *within* the element.

4.5.3 Additive Approximation

The goal stated above is to test whether using the GAI price structure improves the efficiency of a multiattribute auction in comparison to the previously suggested additive price structure. To do that, we should test the performance of the additive price structure given that traders' preferences are not additive, that is they require a GAI structure with some overlapping components. In GAI auctions, the price structure needs only to model the buyer's preferences. If a seller bids on a single configuration in every round, he is not exposed to undesired combinations of values (see Section 4.3.3). The problem is therefore how to select the approximately buyer-preferred sets of configurations, given that the buyer's preferences are not additive. The approach I have taken is to come up with an additive function that approximates the buyer's true utility function, and use it throughout the auction. I do not rule out the possibility that there are better strategies. Note, however, that any such strategy must be consistent: whenever a point is selected as (approximately) optimal, it must remain optimal with respect to the buyer's revealed preferences throughout the auction.

A natural approach to generate a linear approximation $\sum_i \hat{f}_i(\cdot)$ for an arbitrary function $u_b(\cdot)$ is to use linear regression. I define an indicator variable x_{i_j} for any $a_{i_j} \in D(a_i)$, and consider any value of an assignment as a data point. For example, the assignment $a_{1_{j(1)}}, \dots, a_{m_{j(m)}}$ creates the following data point:

$$\sum_{i=1}^m \sum_{a_{i_j} \in D(a_i)} c_{i_j} x_{i_j} = u(a_{1_{j(1)}}, \dots, a_{m_{j(m)}}),$$

in which the value of the variable x_{ij} is 1 if $j = j(i)$ and 0 otherwise. The coefficients c_{ij} result from the regression and represent the values to be used as $\hat{f}_i(a_{ij})$.

When the problem includes a large number of attributes, it is not possible to use all the data points in Θ . Given that the function has a compact GAI representation, meaning a relatively small number of data points can fully describe the function, it is sensible to expect that we could use a smaller number of points for the regression. I therefore tried a regression that models a separate set of data points for each GAI element, meaning it simultaneously performs a regression to the g functions $f_r(\cdot)$ in the GAI decomposition. I found that an auction AP that uses this method achieves lower efficiency compared to auction AP that performs regression over the full joint utility. The reason is that the regression through the compact representation minimizes the error per each sub-configuration, and that may result in larger errors for full configurations. Fortunately, I also found (at least for the sizes of problems I tested) that we can use a small random sample of data points from the joint utility, and that yields an approximation that does as well as an approximation that uses all of the data points of the joint utility. More precisely, for the largest domain I tested (25 attributes, each with domain of size 4) I found that the efficiency of AP does not improve when increasing the number of sampled points beyond 200. I show a chart supporting this claim in the efficiency analysis section, and I use 300 points for all of the experiments.

It is important to note that this method of comparison probably overestimates the quality of an additive approximation. The reason is that typically we will not have the accurate utility function available to us when we generate the approximation. The extraction, or elicitation of the utility function is usually the most serious bottleneck of a multiattribute mechanism. Therefore, the major reason to use an additive approximation is to reduce the burden of elicitation. Hence in practice we will try to obtain the additive function directly, rather than obtain the full utility and then approximate it. The result of such process is somewhat unpredictable, because the elicitation queries may not be coherent: if the willingness to pay for a_1 depends on the value of b , then what is the willingness to pay for a_1 when we do not know b ? I therefore consider this method of approximation a best case of the extent to which an additive approximation may be successful in practice.

4.6 Simulation Results

4.6.1 Measuring Efficiency

All of my efficiency results are measured in terms of percentage of the maximal possible surplus, that is the surplus corresponding to the optimal pair of seller and configuration. The efficiency of the auctions depends on many factors, such as:

1. d , the size of the domains of the attributes (to simplify, I use the same domain size for all the attributes).
2. m , the number of sellers participating.
3. ε , the amount of price decrement.
4. The distribution from which utility functions are drawn.

The goal, however, is to measure the effect of preference modeling on efficiency, and I therefore focus on the parameters reflecting the GAI structure of the buyer's preferences. As shown in the theoretical results, this structure affects efficiency through the factor e . Further, the size of the GAI elements is expected to affect the efficiency obtained by AP. I found that the size of the largest GAI element (denoted by ξ) is particularly crucial for AP. To isolate these factors that I care about, I first describe how the results vary according to the choices of the side factors: the buyer's mean μ_b , the sellers' mean μ_s , σ , d , and m , for several fixed GAI structures with fully random subutility functions (the results of these experiments are provided in Appendix B). This allows me to justify the parameter values I used for the rest of the simulations.

I expect seller's costs to be generally lower than buyer's valuations, otherwise there is no potential surplus. I arbitrarily selected the buyer's mean to be 600, and varied a term I call *the general sellers' mean* from 700 down to 300. Normally, different sellers have different cost levels, so I uniformly draw means for each seller from an interval of size 2σ around the general sellers' mean. I experimented with several values (between 50 and 250) for σ , for both the buyer and the sellers. I found that the choice of these parameters does have a serious effect on the efficiency, particularly the efficiency of AP. The difference between the buyer's mean and the general sellers' mean has similar effect on both AP and GAI (auctions are more efficient when the difference is larger). I picked the value 500 as the general sellers' mean, reflecting costs that are reasonably lower than the buyer's valuation. I also noticed that the smaller the variance is, the better an additive approximation performs. I postulate that it is simply easier to approximate the function when there is less to lose by being wrong. I picked the value $\sigma = 200$, to make sure the problem is sufficiently challenging.

Next, I varied d , between two to ten, for all the attributes. As expected, the larger the domain is, the more challenging the problem is for the additive approximation. Perhaps

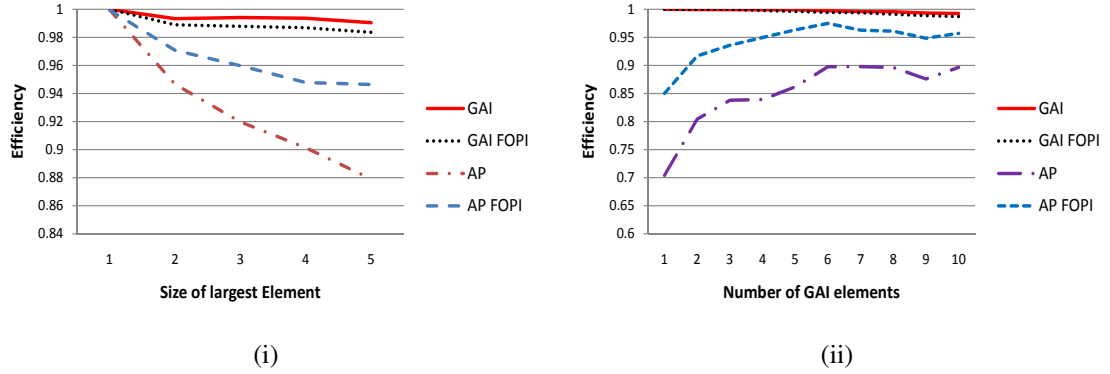


Figure 4.3: Efficiency as a function of: (i) the size of largest GAI element (ξ), given $e = 5$, (ii) the number of GAI elements ($e + 1$), given $\xi = 5$.

less expected is the fact that the size of the domain seems to have no effect on the GAI auctions. I used domain sizes of three to five for the rest of the simulations. As for number of sellers, I varied this parameter from 2 to 40, and did not find a significant effect on either auction types. I used five sellers for the rest of the simulations.

Instead of ϵ , I use the parameter $\delta = \frac{\epsilon}{g}$, which is the actual price decrement per element. I used a few different values for δ , as mentioned below.

4.6.2 Efficiency and GAI Structure

In the next experiment I used a roughly fixed GAI structure, with six elements and $e = 5$ (that is, the GAI network is a tree, not a forest), and $\delta = 4$ ($\epsilon = 24$ in this structure). I vary the number of attributes by varying the size of each element. Figure 4.3i shows the efficiency obtained with respect to ξ , the size of the largest GAI element. As expected, the size of the GAI elements has negligible, or no effect on the efficiency of GAI auctions. It has a dramatic effect on the efficiency of AP. When $\xi = 1$, the decomposition is in fact additive and hence AP performs optimally. The performance then deteriorates as ξ increases.

I performed the same test when using utility in which all attributes are FOPI. Clearly, given FOPI, the additive approximation is much more efficient compared to random utilities. Somewhat surprisingly, the GAI auctions are slightly less efficient given this preference structure, again compared to random utility. Nevertheless, the additive approximation achieves lower efficiency compared to the accurate preference modeling, with differences that pass the statistical significance test ($P < 0.01$), for $\xi \geq 4$. Moreover, in practice FOPI may apply just to a subset of the attributes, in which case it is reasonable to assume that the

efficiency of AP is somewhere between its efficiency under FOPI and its efficiency under random preferences. Also note that the performance of GAI auctions can always be improved using a smaller value of ε and δ , whereas this hardly improves performance of AP. With $\delta = 2$, a statistically significant difference (with the same confidence level) is already detected for $\xi \geq 2$. I used $\delta = 2$ hereafter.

The next experiment (Figure 4.3ii) measures efficiency as a function of e , for a given fixed ξ . I did not test GAI forests, only trees, so e is equivalent to the number of GAI elements minus one. I tested structures with e varying from 1 to 10, all elements of size 3 to 5, and $\xi = 5$ for all the structures.⁵ On a single element, the GAI auction is similar to NLD (Parkes and Kalagnanam, 2005), which is an auction that assigns a price to every point in the joint domain. Here $e = 0$, hence the efficiency of GAI is close to perfect. This structure is on the other extreme compared to an additive representation, and indeed the performance of AP is particularly inferior (only 70% efficient).

With more GAI elements, the efficiency of GAI auctions declines at a very slow pace. The theoretical potential error $(e + 2)\varepsilon$, is mostly a result of efficiency loss of η_i for the winning seller, based on Lemma 4.3. Such efficiency loss may occur only if each sub-configuration in η_i belongs to a configuration that yields the lowest profit allowed in the buyer-preferred set—a particularly rare case. In practice, the loss is closer to $e\delta$, which is a much smaller error.

The performance of AP improves when the number of elements grows while their maximal and average sizes are fixed. The intuitive reason is that changing the structure that way takes it closer to an additive representation. Under FOPI, we see a similar phenomenon as before. However, the difference between GAI FOPI and AP FOPI, even for ten elements, is still significant (and in particular, statistically significant).

Figures 4.4i and 4.4ii show efficiency as a function of the MUI-factor k , for complements and substitutes, respectively. I used a fixed GAI structure with four elements, the largest of which has four attributes, and imposed the same k on all the elements. As expected, the stronger the complementarity is among the attributes, the lower the efficiency of AP, whereas this relationship does not affect the efficiency of GAI auctions. The results are different for the case of substitutes. Here, it seems the additive approximation performs well, and the performance starts to deteriorate only for extreme values of k . Very roughly, we can say that when relationship among attributes (within each GAI element) is limited to (mild) substitutions, it could be a good idea to use an additive approximation. Unfortunately, my interpretation of the parameter k lacks quantitative scaling: there is no clear

⁵I did not find the particular tree structure to be influential on the results; the final structure used in the reported results has a maximum of three children per node.

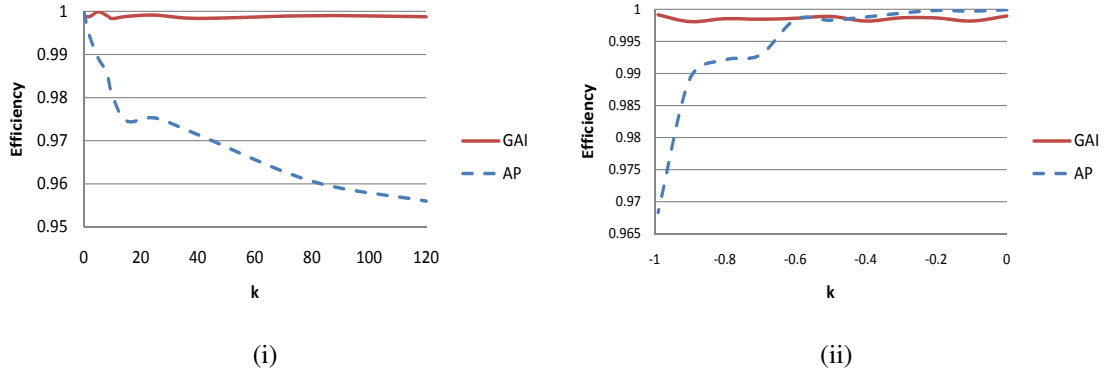


Figure 4.4: (i) Efficiency as a function of $k \geq 0$ (complements). (ii) Efficiency as a function of $k \leq 0$ (substitutes).

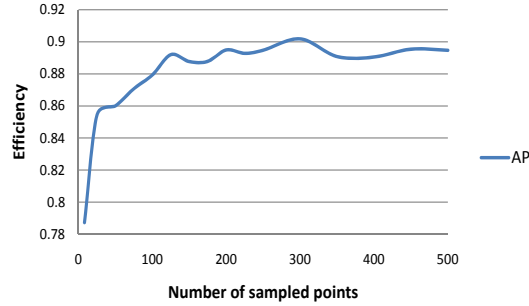


Figure 4.5: Efficiency of AP as a function of the number of sampling points used to devise the additive approximation.

intuition of what the actual numbers mean, beyond the qualitative classification mentioned above.

Finally, I show an experiment supporting the claim in Section 4.5.3: a larger set of sampling points than the one I used for the linear regression of the utility function cannot improve the efficiency of AP. Figure 4.5 shows the efficiency of AP as a function of the number of sampling points used, for the largest domain I used in the experiments: 25 attributes with $d = 4$ ($e = 9$ and $\xi = 5$). Similar results were shown for other distributions and for FOPI preferences. This chart is a result of 150 experiments for each of 10 points on the x-axis, the largest number of tests I used. I found that for a smaller set of experiments the non-monotonicity (exhibited in this chart as well) is aggravated.

4.6.3 Information Revelation

A key difference between the mechanism proposed here and most previous literature is in the extent to which the buyer is required to reveal preference information. In GAI auctions, the buyer does not need to reveal all of its private preference information up front. Of course, the price changes do reveal some of the buyer’s information. Another experimental question is therefore whether this mechanism significantly reduces the overall amount of information revealed by the buyer.

PK study information revelation by both the buyer and the seller, under an additivity assumption. When the utility function is additive the amount of information revealed can be measured as the part of the simplex that is bounded by constraints on the linear weights. Sellers can infer constraints on the buyer’s set of weights, and the amount of information hidden from them is represented by the portion of the simplex that satisfies those constraints. This simplex analysis is not possible for GAI utilities. We suggest an alternative that is geared towards the kind of information revealed by the GAI auctions.

In GAI auctions, the buyer’s private information is partially revealed through the selection of the buyer’s preferred set \mathcal{M}^t . The auction does not need to announce which sub-configurations are in \mathcal{M}^t , so the sellers can infer that a sub-configuration is in \mathcal{M}^t only by observing that it received a bid (usually the will only know that for their own bids), yet its price does not change in the next round. We therefore measure exactly that—for how many sub-configurations θ^r there was at least one round t such that $\theta^r \in \mathcal{M}^t \cap \mathcal{B}_i^t$ for some i . More specifically, we define such a sub-configuration as *revealed*, and within each GAI element we measure the fraction of sub-configurations that are revealed by the end of the auction. This measurement overestimates the information that can actually be inferred by the sellers, because our notion of a revealed sub-configuration provides some relative information regarding the maximal distance of a given configuration from the maximal buyer’s profit, but does not reveal the actual values of the functions $f_b(\cdot)$. Moreover, it assumes that each seller observes all bids (meaning that sellers share bid information with each other) an unrealistic event in practice.

Based on this criteria, GAI auctions reveal on average 15%–25% of the buyer’s preferences when preferences exhibit FOPI, and 10%–15% when the subutilities are completely random. It does not seem to systematically depend on any other parameter I tested. This validates our claim as to the advantage that GAI auctions promise over second-score types of auctions.

4.6.4 Computation and Complexity

The computational tasks required by auction simulations were performed using the algorithms described in Section 4.3.2. These algorithms have been suggested and applied for combinatorial optimization problems before, therefore the computation time is not of particular interest to this work. Instead, I focus on the number of rounds the auction requires. I do however refer to actual time below in order to give a flavor of the overall time complexity.

I tested the number of rounds required by auction GAI and auction AP, under fully random and FOPI preferences. The purpose is twofold: get a sense of the complexity of using GAI auctions, for example with various values of ϵ , and compare it with AP in order to quantify the cost, in terms of number of rounds, required to achieve the higher efficiency provided by GAI auctions. The number of rounds were tested with respect to three of the parameters: ξ (size of largest GAI element), e (connectivity), and δ .

The complexity, in terms of number of rounds, with respect to ξ , is shown in Figure 4.6i. The number of rounds required when $\delta = 2$ is, not surprisingly, roughly twice as many as required when $\delta = 4$. The perhaps surprising result is that for $\xi \geq 5$, the FOPI case requires significantly more rounds to converge than the case of random preferences. Under FOPI, the buyer and all the sellers agree on which attribute values represent high quality and which represent low quality. Therefore, the sellers' and the buyer's preferences can in general be seen as opposites: at the same price, and for a specific attribute, the buyer prefers higher quality, whereas the sellers prefer lower quality (given fixed values for the rest of the attributes). It therefore takes longer for them to converge. The apparent difference in the growth rate (the FOPI case seems to have a steeper curve) is somewhat misleading: for $\xi = 8$ (not shown) GAI under random preferences is already caught up with the same curve we see for the FOPI case. For AP, the only implication of increasing ξ is the respective increase in the number of attributes. It therefore has very mild implication—the complexity of AP (not shown) grows very slowly with the increase in ξ . For the FOPI case, with $\delta = 2$, it takes an average of 481 rounds for $\xi = 1$ (6 attributes) and 546 rounds for $\xi = 6$ (19 attributes). The numbers are slightly higher for random preferences (523 to 628). Whereas AP executes most of the rounds during phase B, it is the opposite case for GAI, which requires a much larger number of rounds in phase A. Phase B in GAI requires a smaller number of rounds than in AP because the sellers have already lost a more significant portion of their profit in phase A.

The results are somewhat surprising in Figure 4.6ii, which shows the number of rounds as a function of e , when $\xi = 5$. Here the difference between the complexity of GAI auctions under FOPI and random preferences seems to be of a qualitative nature. Under FOPI,

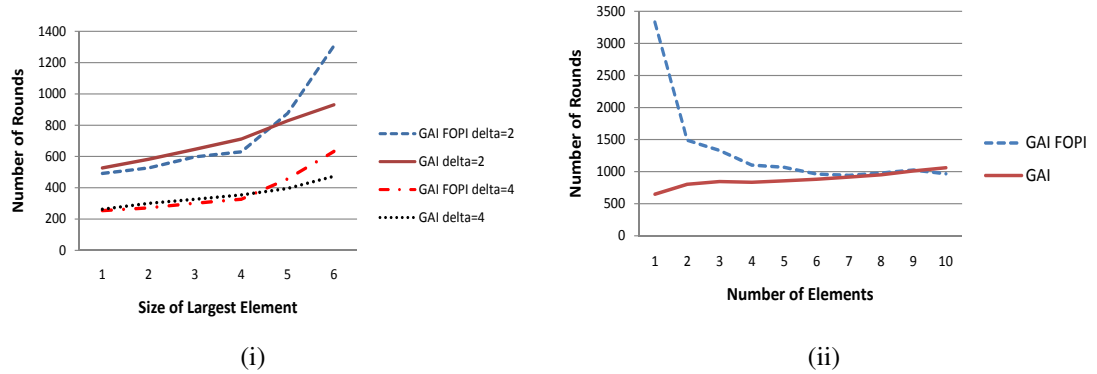


Figure 4.6: Number of rounds as a function of: (i) the size of largest GAI element (ξ), given $e = 5$, (ii) the number of GAI elements ($e + 1$), given $\xi = 5$ and $\delta = 2$.

the number of rounds required for a single GAI element of size 5 is very large (3331 on average) and this number drops quickly when the number of overlapping GAI elements increases, until it converges around $e = 6$. Under random preferences the complexity actually increases slowly with e . The explanation regarding the FOPI phenomenon is that when the number of GAI elements increases, the variance in the buyer's preferences is divided between the elements. With a single element the variance over that element is the largest, and given the opposite preferences of the sellers, it takes many rounds until the buyer's preferred set starts to spread and include more sub-configurations, and therefore it takes longer for phase A to terminate. Under random preferences, this factor is less significant; more salient is the fact that for a smaller e the convergence needs to occur for a smaller number of elements. I found a similar phenomenon for other values of $\xi \geq 3$. However, the overall computation time required by the auction increases with e even for the FOPI case, despite the fact that the number of rounds decreases.

To summarize this part, the GAI auctions require a larger number of rounds to converge, compared to AP. However, on the size of examples I used, the factor is usually below 2. Under FOPI, and a small number of GAI elements, the number of rounds required is fairly large. That might provide an insight as to the complexity of an auction such as NLD (Parkes and Kalagnanam, 2005) that does not use a factorization of the domain. It is also important to note that when we expect the participants of these auctions to use automatic bidding agents, the number of rounds in the ranges I observed in these experiments should not particularly concern. The total computation time, carried out by a GAI auction with 10 elements, $\xi = 5$, $d = 3$, $\delta = 2$, and the rest of the parameters fixed as above, is around 11 seconds on average, using an Intel Dual Core (2.00 Ghz) CPU, with 2048 MB RAM.

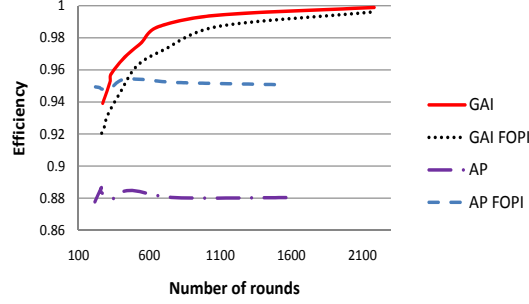


Figure 4.7: Efficiency as a function of the number of rounds.

4.6.5 Comparison for a Fixed Number of Rounds

I expect bidding in high dimensional multiattribute auctions to usually employ automatic bidding agents, each equipped with the cost function of the seller on behalf of which it bids. Under these circumstances, as mentioned, we need not worry about the auction taking up to thousands of rounds. However, if for some reason rounds are expensive, we might need to reconsider using the additive auctions, and sacrifice efficiency in order to decrease the number of rounds. Alternatively, we could keep using GAI auctions and increase δ . The final experiment compares these two alternatives. In this experiment I vary the level of δ , in order to view the efficiency as a function of the number of rounds (Figure 4.7). The structure used for this experiment had $e = 5$ and $\xi = 5$.

As evident from the chart, in most cases GAI achieves better efficiency even for a fixed number of rounds. The only exception is when the budget of rounds is very small (under 200), and FOPI holds. In such case we need to pay with more rounds in order to get the higher efficiency.

4.7 Summary

I propose a novel exploitation of preference structure in multiattribute auctions. Rather than assuming full additivity, or no structure at all, I model preferences using the GAI decomposition. I developed an iterative auction mechanism directly relying on the decomposition, based on the results of Chapter 3 which provide direct means of constructing the representation from relatively simple statements of willingness-to-pay. My auction mechanism generalizes PK's preference modeling, while in essence retaining their information revelation properties. It allows for a range of tradeoffs between the expressiveness of the buyer's preferences and both the complexity of the pricing structure and efficiency of the

auction, as well as tradeoffs between buyer's information revelation and the time required for convergence.

I performed a simulation study of my proposed multiattribute auctions, compared to a mechanism that assumes additive preferences. The study validated the usefulness of GAI auctions when preferences are non-additive but GAI, and allowed me to quantify the advantages for specific classes of preferences. In particular, I found significant benefit to supporting the accurate preference structure, especially when the GAI subutilities do not exhibit FOPI. When the local functions do exhibit structure of their own, in most cases the benefit of an accurate GAI model is still significant. Using an additive approximation may be a reasonable approach when the GAI structure is fairly similar to an additive one, or when attributes within each GAI element are restricted to be substitutes.

I believe this study provides several methodological lessons applicable to a broader class of preference research problems. First, I investigated the problem of generating structured random utility functions. These functions are random to the extent that they conform to a specific GAI structure, and that the preference order over each element exhibit a structure of our choice: FOPI, or MUI with some predetermined level of complementarity or substitutivity. Second, I studied the problem of finding an additive approximation to an arbitrary GAI function. I found that performing linear regression using a relatively small set of random points achieves an approximation that does as well as one performed using all the points in the domain.

Chapter 5

Multi-Unit Multiattribute Auctions

5.1 Introduction

In this chapter I investigate problems associated with multi-unit multiattribute auctions. I use this term to refer to auctions in which the quantity is not treated as a regular attribute. Rather, the quantity desired by a trader can be divided in some way between different trading partners, resulting in multiple trades for a given trader. Single-unit auctions refer to auctions that are restricted to allocate a single trade per trader, but quantity may still be supported as a regular attribute.

Perhaps due to the complexity of multiattribute auctions, nearly all prior proposals for preference handling and mechanism design for multiattribute procurement auctions are restricted to the single-unit case, also called *single-sourcing*, meaning that a single bidder is selected as supplier. As single-supplier relationships introduce a risk of cost overruns and supply disruptions or stock-outs, buyers often have a preference for *multi-sourcing*, where supply contracts are distributed among multiple winners. Limited supplier capacities may also contribute to the need for multi-sourcing. In practice, many procurement auctions have dealt with multiple units over multiattribute goods (e.g., (Metti et al., 2005; Sandholm et al., 2006)). To my knowledge, however, these do not support structured multiattribute preferences or theoretical properties regarding the efficiency across the space of configurations.

To address multi-sourcing issues, I consider *multi-unit multiattribute (MUMA)* auctions. Due to the inherent complexity of this problem, I first consider the representation of preferences over this domain in isolation, and later extend the focus to mechanism design and algorithmic problems. The MUMA problem combines the dimensionality of the single-unit multiattribute domain with the combinatorial complexity of multi-unit auctions of heterogeneous goods. I consider the problem in its full generality, then suggest simplifications based on assumptions over preference structure.

Some issues relevant to my work are explored in the literature on *side constraints*, which place hard constraints on the space of allocations acceptable to the bid taker in multi-object and combinatorial auctions. Sandholm and Suri (2001) show that most such constraints posed by the bid taker render the winner determination problem NP-hard. Bichler and Kalagnanam (2005) explicitly consider the problem of multi-unit multiattribute procurement auctions, focusing on the winner determination problem given various buyer-imposed constraints. My approach is a general framework to express *preferences* over the multi-unit multiattribute domain rather than hard constraints. In that sense another relevant work is that of Boutilier et al. (2004b), which opts for a utility-based representation over allocation attributes in combinatorial auctions, and focuses on the elicitation of tradeoffs between constraints and cost.

The preference representation perspective brings several advantages. It captures a fully general representation of this domain, a capability that previous literature lacked. It lends itself easily to the expression of the welfare maximization problem, and most importantly allows the use of well-founded tools from multiattribute utility theory to simplify the preference representation. This framework lets me use meaningful preference structure in order to achieve value decomposition, which I apply to effectively decouple the multiattribute domain from the multi-unit problem. I then leverage this result to create an auction mechanism for multi-sourcing procurement.

The second part of this chapter considers another major generalization: two-sided auctions, that is an exchange involving multiple buyers and multiple sellers. A two-sided auction is an economic tool that can yield higher efficiency by aggregating buyers and sellers in a central trading institution. I study the problem of clearing a two-sided multi-unit multiattribute call market. Here, I address the algorithmic problem but not the strategic issues. In particular, I reason directly from bid expressiveness to the computational complexity of clearing, finding constraints on the bidding language which lead to tractable algorithms.

5.2 MUMA preferences

5.2.1 The Global Multiattribute Allocation Problem

Given that the scope of the chapter include the two-sided auction settings, I present the preference framework for the general settings of two-sided auctions. I first lay out the *global* multiattribute allocation problem, in which a set of buyers B and a set of sellers S wish to

trade amounts of a particular multiattribute good. For example, in an exchange between car dealers, each has various preferences on the mix of cars they wish to add to their lot. The multi-unit procurement problem is of course a special case of this setting in which $|B| = 1$.

When allowing multiple trades for a single trader at the same time, we need to consider traders' preferences over several issues that do not come up in single-unit allocations:

1. How many trading partners would a trader prefer to trade with? For example, a procurement department may wish to distribute the purchased quantity over at least two suppliers, but also to limit that number to four. Another common preference is over the distribution of the *expenditure* rather than the quantity.
2. Must all trades be on the exact same configuration? Examples can be given of cases in which configurations should not vary over the trades, but also to opposite cases in which companies would prefer to trade a mixture of different configurations. Often, a trader would be willing to trade different configurations, as long as some specific attributes are restricted to be *homogenous*, that is carry the same value in all the trades (Bichler and Kalagnanam, 2005).
3. How to treat preferences over quantities? In many cases, preferences are linear in quantity (at least within some bounded range), but not always. Suppliers, for example, may have nonlinear preferences over the quantity they supply, due to fixed costs associated with production.

Initially, I present a framework that allows for fully general preference expression over these issues. Later, I restrict specific components in order to achieve economic and algorithmic results. To simplify the presentation, I refer to a buyer's preferences in forming the definitions and claims, but they all apply to sellers with only minor modification. In Sections 5.4 and 5.5 I present an auction mechanism which is restricted to the procurement case, before returning to the exchange setting in Section 5.6.

For any global set of trades $T \subset \mathcal{T}$ (see notations in Section 2.3.1), I denote the allocation to buyer \hat{b} by $T_{\hat{b}}$,

$$T_{\hat{b}} = \{(\theta, q, b, s) \in T \mid b = \hat{b}\}.$$

Note that

$$\forall T, \bigcup_{b \in B} T_b = \bigcup_{s \in S} T_s = T. \quad (5.1)$$

Let $\mathcal{T}_{\hat{b}}$ denote the set of all possible trades in which \hat{b} participates:

$$\mathcal{T}_{\hat{b}} = \{(\theta, q, b, s) \in \mathcal{T} \mid b = \hat{b}\},$$

and let $\mathcal{O}_{\hat{b}}$ denote the set of all possible allocations to \hat{b} :

$$\mathcal{O}_{\hat{b}} = \{T_{\hat{b}} \mid T \subset \mathcal{T}\} = 2^{\mathcal{T}_{\hat{b}}}.$$

Any element of $T_{\hat{b}} \in \mathcal{O}_{\hat{b}}$ can be described using a non-negative vector of length $|\Theta||S|$, where each component designates the number of units that \hat{b} buys of a specific (θ, s) pair (where zero quantity reflects no trade). As in other parts of the work, I assume that \hat{b} has a full preference order over $\mathcal{O}_{\hat{b}}$, the set of possible outcomes to \hat{b} .

I adopt the following common assumptions from mechanism design:

1. *free disposal*, meaning that the utility of \hat{b} can only increase when an item is added to $T_{\hat{b}}$.
2. The utility of \hat{b} is not affected by the allocations T_{τ} for any trader $\tau \neq \hat{b}$.

Under these assumptions, and as described in Section 2.1.1, \hat{b} 's preferences can be represented by a utility function (MVF) $u_{\hat{b}} : \mathcal{O}_{\hat{b}} \rightarrow \mathfrak{R}$. As described in Section 3.3, $u_{\hat{b}}$ can be scaled to monetary terms, and represent willingness-to-pay per allocation. In the case of a seller s , the utility $u_s(\cdot)$ is usually referred to as cost, and represented in bidding as willingness-to-accept. I use the term WTP in referring to both buyers and sellers.¹

I define the welfare optimization problem for this domain:

Definition 5.1. *The global multiattribute allocation problem (GMAP) is defined as follows:*

$$GMAP = \max_{T \subseteq \mathcal{T}} \left(\sum_{b \in B} u_b(T_b) - \sum_{s \in S} u_s(T_s) \right)$$

Note that *MAP* is a special case of *GMAP* in which $|B| = 1$ in the input, and $|T| = 1$ is a constraint on the result.

5.2.2 Preference Structures

The first restrictive assumption I make is that a trader can trade only a single configuration with each trading partner. Though non-trivial, I find this assumption reasonable for this domain. Unlike combinatorial auctions, the primary goal of multiattribute auctions is to trade on surplus-maximizing *configurations*, rather than a surplus-maximizing *combination* of goods. From a buyer's perspective, the primary goal of the multi-unit generalization is to achieve an ideal supplier mix, while achieving a heterogeneous configuration mix is at best secondary. The heterogeneity of supply *from the same supplier*, if desired at all, is even lower in priority. On a seller side, there is also very little motivation to prefer a mix of configurations with the same buyer.

This assumption leads to an exponential reduction in dimensionality: any allocation can now be defined by a vector of length $(n+1)|S|$ (note $|A| = n$), which I represent as a matrix

¹I use u_j when referring to a seller s_j , and drop the subscript altogether in generic statements about utility, when not referring to a particular trader.

of $n + 1$ columns (one for each attribute, and one for the quantity of the trade) and $|S|$ rows (one per trading partner). I call this matrix an *allocation matrix*, because an instance of this matrix represents an allocation. For example, in buyer \hat{b} 's matrix, row j represents a trade between \hat{b} and seller s_j , where the first n columns define the trade configuration (as values to attributes a_{1j}, \dots, a_{nj}), and the attribute at the last column, q_j , defines the trade quantity. The term “attributes” used here includes the $(n + 1)|S|$ elements of the matrix, that is the attributes of the multi-unit domain. It is important to stress that an instantiation of this matrix represents a single allocation, and that expressing WTP over all allocations requires the definition of a reserve price for every possible such instantiation.

Even with this significant simplification, the domain is still too large. The representation of utility over the allocation matrix is at best exponential in the number of elements of the matrix (assuming all attribute domains are discretized and bounded). Given such a representation we cannot expect computational mechanisms to solve *GMAP* for more than tiny instances. However, since the domain is now representable by a restricted set of attributes, we can use tools from multiattribute utility theory to tame it further. In particular, I use the GAI representation scheme (Section 3.2). Recall the following functional structure from Equation (3.5):

$$\begin{aligned} f_1 &= u([I_1]), \text{ and} \\ \text{for } r = 2, \dots, g, \quad f_r &= u([I_r]) + \sum_{j=1}^{r-1} (-1)^j \sum_{1 \leq i_1 < \dots < i_j < r} u\left(\bigcap_{s=1}^j I_{i_s} \cap I_r\right). \end{aligned} \tag{5.2}$$

Though complicated on the surface, this expression bears intuitive meaning. We simply add up utilities of subsets, but to avoid double counting we subtract utilities of pairwise intersections, at which point we have over-deducted and must add back those of three-wise intersections, and so on. I explicitly apply this functional form in some of the results below.

I denote the attributes that refer to the trade with s_j by $t_j = \{a_{1j}, \dots, a_{nj}, q_j\}$ (the set containing the attributes in row j), and denote the set of attributes in the quantity column of the matrix by $\vec{q} = \{q_1, \dots, q_{|S|}\}$.

The functional form above relies on predefined reference values for all attributes. For the MUMA domain I define zero as the reference value for each attribute $q_j \in \vec{q}$. For the remaining attributes a_{ij} I use constants \tilde{a}_i , which comprise the reference configuration $\tilde{\theta} = (\tilde{a}_1, \dots, \tilde{a}_n)$. This means for example that $u_b([t_j])$ corresponds to the WTP of b for the trade (θ_j, q_j, b, s_j) , as if it is the only trade in the allocation to b , and that $u_b([q_j])$ is the WTP for q_j units of $\tilde{\theta}$, again as the only trade in the allocation. Based on this notion, I for-

mally introduce preferences over a single-unit allocation, which I designate by the utility (WTP) function $\mu_b : \Theta \times S \rightarrow \Re^+$:

$$\mu_b(\hat{\theta}, s_j) = u_b([\{a_{1_j}, \dots, a_{n_j}, q_j\}]), \quad (a_{1_j}, \dots, a_{n_j}) = \hat{\theta}, q_j = 1,$$

referring to the WTP of b for an allocation that contains a single-unit trade between b and s_j over configuration $\hat{\theta}$. The single-unit utility of sellers is defined similarly using the notation μ_s , or μ_j when referring to a particular seller s_j .

The single-unit utility function is a multiattribute WTP function, used in previous sections, with the minor generalization: it admits an additional argument for the identity of the trading partner. This is just a minor generalization, because it could be accommodated before as an additional attribute in A . The *MMP* is therefore the same problem as presented in Definition 2.18, denoted here as follows:

$$MMP(b, s) = \arg \max_{\theta \in \Theta} (\mu_b(\theta, s) - \mu_s(\theta, b)).$$

The next step is to define several restrictive classes of traders' preferences.

Definition 5.2. *A buyer b is called configuration aggregating if her preferences exhibit GAI structure over the following collection of subsets:*

$$\{t_1, t_2, \dots, t_{|S|}, \vec{q}\}.$$

In words, a configuration aggregating buyer has additive utility over different trades, subject to preferences over distribution of quantity. Though this rules out any constraints on aggregation of configurations, it does not rule out various allocation preferences as explained below. Note that the collection of GAI elements in the above definition is disjoint, with the exception of $t_j \cap \vec{q} = \{q_j\}$. Therefore when applying the functional form in Equation (5.2) we get a summation over all elements minus those pairwise intersections:

$$u_b(T_b) = \sum_{j=1}^{|S|} u_b([t_j]) + u_b([\vec{q}]) - \sum_{j=1}^{|S|} u_b([q_j]). \quad (5.3)$$

The second and third terms above together are what I call the “quantity factor”, which is the utility factor of buyer b over the GAI element \vec{q} , denoted by $f_b^q(\cdot)$:

$$f_b^q(\vec{q}) = u_b([\vec{q}]) - \sum_{j=1}^{|S|} u_b([q_j]). \quad (5.4)$$

The interpretation of the quantity factor is that it expresses utility over the *combinations* of trades from different trading partners, whereas the factors $u_b([t_j])$ represent utility of each trade as an allocation by itself. Section 5.3.1 provides a full exposition of this concept.

Definition 5.3. *Buyer b 's preferences exhibit linear trade utility if there exist q^b and \bar{q}^b such that*

$$u_b([t_j]) = g_b(q_j)\mu_b(\theta, s_j) \quad \text{for } j = 1, \dots, |S|,$$

and

$$g_b(q_j) = \begin{cases} 0 & q_j < q^b \\ q_j & q^b \leq q_j \leq \bar{q}^b \\ \bar{q}^b & q_j > \bar{q}^b. \end{cases}$$

The bounds q^b and \bar{q}^b are called minimal and maximal trade quantity, respectively. Similarly q^j and \bar{q}^j denote the bounds for s_j .

In words, the utility of each single trade (as the only trade in the allocation) is linear in quantity, subject to strict trader-specified upper and lower bounds. In the case of sellers, the utility (cost) below the lower bound equals the cost at the lower bound, and cost above the upper bound is considered infinite. Although *linear trade utility* is a hard constraint (required to achieve the result of Theorem 5.1), it does not compromise the flexibility of the quantity factor $f_b^q(\vec{q})$.

5.2.3 MMP-GMAP decomposition

The combination of linearity in trade quantity and configuration aggregation leads to a significantly simplifying result.

Theorem 5.1. *1. If all traders exhibit configuration aggregation and linear trade utility, then there exists a solution to GMAP that consists of a set of trades, each of which is on a configuration that is a solution to MMP for its specified pair of traders.*
2. For any sets of traders B and S (and either $|B| > 1$ or $|S| > 1$ or both), if the preference structure of at least one of the traders τ does not agree with configuration aggregation, then for any utility functions for the rest of the traders, there exists a utility function $u_\tau(\cdot)$ such that any solution to GMAP includes a trade on a non-MMP configuration.

To prove the first claim, I first show, based on the definition of the quantity bounds and free disposal, that a trade can occur only within the quantity bounds of all traders (or can be replaced with one that does). We can therefore assume $g_b(q_j) = q_j$, and then the following

functional decomposition is implied by the conditions above:

$$u_b(T_b) = \sum_{j=1}^{|S|} q_j \mu_b(\theta_j, s_j) + f_b^q(\vec{q}), \quad (5.5)$$

and similarly for sellers. It is then easy to show that improving any bilateral solution would improve the total surplus of *GMAP*.

The usefulness of this decomposition is in the fact that the global optimization problem no longer depends on the attributes, because the attributes need to be considered only in the *MMP* stage. It effectively separates the problem to a multiattribute problem, and a price-only multi-unit problem. This opens the way to a procurement mechanism, and a generalized network flow algorithm for *GMAP*, both introduced later in this chapter. The conditions of configuration aggregating and linear trade utility are fairly restrictive, however they still allow a wide variety of expressive preferences, as described next.

5.3 Aggregation Preferences

5.3.1 Aggregation over Trading Partners

Theorem 5.1 generalizes a previous result (Engel et al., 2006) by accommodating arbitrary functions over the vector of quantities, thus enabling aggregation preferences over trading partners. For example, a buyer may want to reduce the number of suppliers she deals with, but finds a hard constraint (e.g., at least x suppliers) overly restrictive, since she might want to balance her preference for more suppliers against the cost of including inefficient suppliers. The trade elements cannot express such preferences, but the quantity factor $f_b^q(\cdot)$ can vary the total WTP based on the distribution of quantity.

Consider the form of the quantity factor in Equation (5.4). This factor can increase the WTP for allocations in which multiple elements of \vec{q} are non-zero. That happens when the utility of all the trade quantities taken together (or the *utility of the quantity distribution*), $u_b([\vec{q}])$, is greater than the sum of the individual trade quantity utilities, $u_b([q_j])$. Similarly, preferences towards a small number of suppliers result in $u_b([\vec{q}]) < \sum_{j=1}^{|S|} u_b([q_j])$ for allocations that have too many suppliers, meaning that the quantity factor penalizes the total WTP. Note that a different choice of reference values (non-zero values for \tilde{q}_j) would not interfere with this interpretation since the non-relevant units cancel out in Equation (5.3).

For example, assume a buyer \hat{b} wishes to buy up to six units, and prefers to divide the quantity evenly among two sellers. In such case the value of $u_{\hat{b}}([\vec{q}])$ is highest for \vec{q} that

have exactly two equal non-zero elements, lower the more uneven the distribution is, and also lower for vectors with more than two non-zero elements. The term $\sum_{j=1}^{|S|} u_b([q_j])$, on the other hand, does not depend on the distribution of quantity. Therefore $f_b^q(\cdot)$ modifies the WTP according to how close is the allocation to \hat{b} 's quantity distribution preference. More generally, any configuration-independent preferences over the distribution of the allocation to suppliers can be generated by the quantity factor. These preferences cannot depend on specific configurations, because that would violate the GAI structure. For example, a configuration aggregating buyer cannot express a preference such as “increase the number of suppliers when the quality of insurance is low”. Such a statement can be expressed using another GAI element, which includes \vec{q} and the column of the “insurance” attribute. Note that better insurance per se is rewarded through the configuration utility factors $u_b([t_j])$.

As mentioned, the definitions and analysis apply to sellers with minor adjustments. It is important to note however that when there is a single seller, $f_b^q(\vec{q}) = 0$ according to (5.4). Similarly, in procurement mechanisms with a single buyer there are no quantity factors for sellers.

To illustrate the expressiveness of the model and its limitation, I consider the following problem. A buyer \hat{b} represents a procurement department that wishes to purchase LCD monitors for the company's employees. Negotiable attributes include contractual aspects such as warranty, delivery, insurance of deal, and payment terms. Suppliers are also evaluated on a measure of product quality and supplier reliability reputation, data which is available to \hat{b} . Each configuration also provides product specifications over resolution, number of colors, number of nits, and its compatibility with wall mounting hardware.

It is reasonable to assume that \hat{b} does not object to having suppliers with different contractual attributes and different quality and reliability ratings, as long as the payment for each supplier reflects those values. It is also likely that interdependencies among the attributes exist. For example, the WTP for longer warranty can depend on the product quality. These kind of interactions at the multiattribute level are not ruled out by the model. \hat{b} can also express preferences over the number of suppliers and the distribution of quantities using the quantity factor. The restrictions of Theorem 5.1 begin to arise if \hat{b} has particular preferences on the mix of qualities. For example, \hat{b} might wish to have some percentage of higher-end monitors (higher resolution, # colors, etc...) and some of lower-end, cheaper ones, for different types of employees. Though not directly supported, \hat{b} could specify higher-end suppliers and lower-end suppliers and consider only a partial domain for each. In addition, \hat{b} might prefer that all suppliers have the same value for wall-mounting compatibility, so \hat{b} will not need to purchase different kinds of such hardware. These considerations violate configuration aggregation (Definition 5.2) and hence cannot be ac-

commodated by mechanisms designed according to Theorem 5.1. However, they can still be expressed by the preference representation scheme, as detailed in the next subsection.

5.3.2 Restricting Configuration Aggregation

Configuration aggregation, required by Theorem 5.1, prevents traders from expressing any constraints or preferences on the mix of configurations in their allocation. More specifically, if a trader is aggregating she must be willing to trade any combination of configurations, at prices defined by her WTP per configuration and trading partner. This is quite restrictive, since traders may wish to limit the variance of specific attributes (for example, “items from all suppliers must be delivered at the same time”, or the compatibility issue in the monitors example). Bichler and Kalagnanam (2005) introduced such preferences as *homogeneity constraints*, treating them as hard constraints on acceptable allocations.

Another, more expressive way to describe these types of preferences is to say that configurations have marginal utilities which vary based on the allocation in which they are traded. This can be expressed in the GAI framework by expressing a penalty term which includes the relevant attributes. For example, a trader might have a preference against the aggregation of units of different colors. Assume that c represents the attribute color (with values c^1, c^2), and a_1, \dots, a_{n-1} represent the rest of the attributes. For simplicity assume there are only two potential sellers, s_1, s_2 . The preference for homogeneity indicates the following:

$$\forall q'_1, q'_2, u([c_1^1, c_2^2, q'_1, q'_2]) < u([c_1^1, q'_1]) + u([c_2^2, q'_2])$$

That means that the utility for an allocation with two configurations of different colors is less than the combined utility of two allocations, each includes only one of the colors. To extend the GAI representation of an aggregating trader to support such expressiveness, we thus have to add another GAI element, $\hat{c} = \{c_1, c_2, q_1, q_2\}$. Since the two utility terms on the RHS above are defined over $\hat{c} \cap t_1$ and $\hat{c} \cap t_2$ respectively, the explicit GAI decomposition from (5.2) lends itself easily to expression of the penalty:

$$u(T_b) = u([t_1]) + u([t_2]) + u([c_1^1, c_2^2, q'_1, q'_2]) - u([c_1^1, q'_1]) - u([c_2^2, q'_2])$$

In this model, the utility of allocation is an additive sum of the utility of trades, as long as the colors are the same.

The key lesson from this discussion is that this preference handling approach—using GAI decomposition over the allocation matrix—flexibly adds complexity only as required

by traders' preferences. This framework avoids specifying hard constraints on allocations, and allows for expression of specific non-regularities by adding GAI elements to the model. For the specific conditions of Theorem 5.1, this framework leads to useful computational and economic results as I demonstrate in the next section.

5.4 MUMA GAI Auctions

With the decomposition guaranteed by Theorem 5.1, it seems natural to design a two-phased mechanism for the multi-unit multiattribute domain: a single-unit multiattribute mechanism to solve *MMP* for all pairs of traders, followed by a multi-unit price-only auction to determine the quantity distribution. In this section I propose such a mechanism for the (one-sided) procurement problem. The mechanism relies on the single-unit GAI auctions introduced in Chapter 4: in essence, we use the same Phase A and generalize Phase B to multi-unit.

Assume traders have MUMA preferences that satisfy the conditions of Theorem 5.1. In addition, we retain the GAI structure $I_1, \dots, I_g \subseteq A$ of the single-unit function $\mu(\cdot)$ as in Chapter 4, and it is independent of the preference structure of the allocation matrix imposed by Definition 5.2.

I define MUMA GAI auctions as follows. The buyer initially reports her valuation under the structure of Theorem 5.1, which can be represented using a single-unit valuation function $\mu(\cdot)$ (decomposed by its GAI structure), trade quantity bounds \underline{q}_b and \bar{q}_b , and a quantity factor $f_b^q(\bar{q})$. We then execute Phase A, identical to the one defined for the single-unit case. As mentioned earlier, the notion of *MMP* is identical in the single-unit and multi-unit frameworks, and therefore from Lemma 4.8 we get that the configurations η_j are approximate *MMP* solutions for each seller s_j .

In the single-unit version, Phase B is a price-only ascending clock auction (the discount term corresponding to a price). Similarly, for Phase B in the multi-unit case we use a multi-unit (price-only) auction. For the purpose of the analysis below, I use a direct revelation VCG mechanism, in which each seller s_j reveals $\mu_j(\eta_j)$ and \bar{q}_j (recall that a seller's quantity factor must be zero when there is only one buyer, and that we require $\underline{q}_j = 0$).

Theorem 5.2. *Under the preference restriction of Theorem 5.1, and assuming SB sellers and a truthful buyer, a MUMA GAI auction which employs an efficient (price-only) mechanism as Phase B, achieves total surplus which is within $q\epsilon(e+1)$ of the efficient allocation, where q is the number of units allocated by the efficient solution.*

If (instead of VCG) we use in Phase B a mechanism that guarantees only an approximate solution within some δ of optimum, the theorem above and its proof can be easily adapted to show that the auction would achieve surplus within $\delta + q\epsilon(e + 1)$ of optimum. For example, the mechanism we apply in the special case of single-unit (Chapter 4) is ϵ -efficient, resulting in the overall approximate efficiency of Theorem 4.10.

The incentive properties of the auction can be established based on those of Phase B.

Theorem 5.3. *Given a truthful buyer, if the mechanism used in Phase B yields sell-side VCG prices (with respect to its input), then SB is an ex-post $2q\epsilon(e + 1)$ -Nash Equilibrium for sellers in the MUMA GAI auction. That is, sellers cannot gain more than $2q\epsilon(e + 1)$ by unilaterally deviating from SB.*

As with the previous result, this theorem can be adapted to approximate VCG mechanisms, adjusting the potential gain accordingly. Since these results assume nothing about the Phase B mechanism other than efficiency and VCG prices, they apply to any efficient iterative mechanism implementing the VCG outcome. A direct generalization of the single-unit mechanism would be an iterative forward auction over a discount from Phase A prices, (e.g., Ausubel (2004)). However, the problem of non-convex utility on the bid taker side (buyer in our case), reflected in $f_b^q(\cdot)$, is rarely addressed in the context of iterative mechanisms. We therefore might have to resort to a direct revelation VCG mechanism when a non-trivial buyer quantity factor $f_b^q(\cdot)$ exists. Indeed, the computational problem associated with the *WDP* of the VCG mechanism can be intractable, but several useful cases of non-convex utility can be handled using generalized network flow algorithms explored later in this chapter. These algorithms are later shown to perform reasonably well in clearing a two-sided market with up to hundreds of traders.

There are other, well-known drawbacks to a direct revelation approach, for example in the communication burden of full evaluation. I therefore stress that the advantage achieved with this mechanism is the reduction of the multiattribute problem to a price-only problem, by the means of using Phase A and Theorem 5.1. In practice, procurement problems rarely involve more than a few potential suppliers, and the quantity can be discretized to keep the total number of units low. In such case the *WDP* of Phase B is tractable, even though the attributes domain can be extremely large. Again I refer to the KLM buttercup procurement example. The problem had 77 attributes (as mentioned earlier), but only seven potential suppliers, two of which were pre-eliminated.² Therefore, the combination of compact representation for the single-unit preferences and the decoupling of the multiattribute domain from the multi-unit problem can render the problem solvable.

²Source: <http://www.negometrix.com>

5.5 Example

For this example, the buying agent \hat{b} procures a large quantity of hard drives for internal use. \hat{b} is willing to accumulate hard drives with various characteristics, as long as the payment varies accordingly. Specifically, there are three attributes, with two possible values each:

Capacity (c) $c^1 = 60GB, c^2 = 120GB$.

RPM (r) $r^1 = 3600rpm, r^2 = 5400rpm$.

Warranty (w) $w^1 = 3mths, w^2 = 6mths$.

The GAI structure is $I_1 = \{r, c\}, I_2 = \{c, w\}$ (note that $e = 1$). The resulting sub-configurations are therefore $r^1c^1, r^2c^1, r^1c^2, r^2c^2$, and $c^1w^1, c^2w^1, c^1w^2, c^2w^2$.

	I_1				I_2			
	r^1c^1	r^2c^1	r^1c^2	r^2c^2	c^1w^1	c^2w^1	c^1w^2	c^2w^2
$\mu_{\hat{b}}$	60	85	70	95	40	60	60	70
μ_1	50	60	55	75	25	30	40	50
μ_2	35	45	65	90	20	40	30	50

Table 5.1: GAI utility functions for the example domain. $\mu_{\hat{b}}$ represents the \hat{b} 's valuation, and μ_1 and μ_2 costs of s_1 and s_2 .

The procurement department of \hat{b} considers two potential suppliers, s_1 and s_2 . Since the company wishes, for future reasons, to maintain relationship with both, it has some preference towards dividing the quantity between the two suppliers. For simplicity I divide the large quantity to two equal units, meaning $\bar{q}_{\hat{b}} = 2$. All traders' utilities exhibit configuration aggregation and linear configuration utility. The sellers' lower quantity bound is zero, and their capacity is unlimited in the relevant range (meaning $\bar{q}_1, \bar{q}_2 \geq 2$).

The buyer's and the sellers' single-unit utility functions, $\mu_{\hat{b}}, \mu_1, \mu_2$ are given in Table 5.1. The buyer's preference towards distribution of supply is expressed through the quantity factor $f_{\hat{b}}^q$ of her utility as follows:

$$f_{\hat{b}}^q(q_1, q_2) = \begin{cases} 30 & q_1 = q_2 = 1 \\ 0 & 2 \geq q_i > 0, q_j = 0 \ (i \neq j \in \{1, 2\}) \end{cases}$$

I first analyze Phase A, whose progress is given in Table 5.2. Let $\varepsilon = 8$, so the price decrement per sub-configuration is 4. The efficient allocation is $(s_2, r^2c^1w^2)$, yielding a

	I_1				I_2			
	r^1c^1	r^2c^1	r^1c^2	r^2c^2	c^1w^1	c^2w^1	c^1w^2	c^2w^2
1	100	100	100	100	75	75	75	75
	s_1, s_2			*	s_1, s_2			*
2	96	100	100	100	71	75	75	75
	s_1, s_2			*	s_1, s_2			*
3	92	100	100	100	67	75	75	75
	s_2		s_1	*	s_2	s_1		*
4	88	100	96	100	63	71	75	75
		s_2	s_1	*		$s_1, *$	s_2	*
5	88	96	92	100	63	71	71	75
	s_2		s_1	*	s_2	$s_1, *$		*
6	84	96	88	100	59	71	71	75
		s_2	s_1	*		$s_1, *$	s_2	*
7	84	92	84	100	59	71	67	75
	s_2	*	s_1	*	s_2	$s_1, *$	*	*
8	80	92	80	100	55	71	67	75
		$s_2, *$	$s_1, *$	$s_1, *$		$s_1, *$	$s_2, *$	*

Table 5.2: Auction progression in phase A. Sell bids are denoted by s_1, s_2 and designation of \mathcal{M}^t by $*$.

surplus of $145 - 75 = 70$. s_1 's efficient configuration is $r^2c^2w^1$ with surplus of 50. We set high initial prices, 100 for all sub-configurations in I_1 and 75 for I_2 .

Initially, prices of all configurations are equal, causing the sellers to bid on their cheapest configuration ($r^1c^1w^1$). The buyer's preferred configuration is his most valuable one ($r^2c^2w^2$). s_1 changes his bid to $r^1c^2w^1$ in round 3, since its current profit is $175 - 85 = 90$, compared to $159 - 75 = 84$ of $r^1c^1w^1$. This configuration remains optimal until round 8 (though the profit is reduced to $151 - 85 = 66$), in which $r^2c^2w^1$ becomes optimal too, with the same profit. s_2 remains on $r^1c^1w^1$ until round 4, and alternates between this and $r^2c^1w^2$ at each round. In round 7, $r^2c^1w^2$ finally becomes buyer preferred (with buyer profit of -14) and in the next round, when s_2 bids $r^2c^1w^2$ again, both sellers are bidding on configurations that are in the buyer's preferred set. Phase A therefore terminates with $\eta_1 = r^2c^2w^1$ (which is chosen over $r^1c^2w^1$ since it yields higher buyer profit) and $\eta_2 = r^2c^1w^2$. If we had been using an iterative mechanism for phase B, the sellers would compete on a discount from the prices of round 8, and given their profit margins $\pi_1(\eta_1) = 171 - 105 = 66$ and $\pi_2(\eta_2) = 159 - 75 = 84$, s_1 would have dropped out once the discount exceeded 66.

However to take $f_b^q(\cdot)$ into account, we use a direct VCG mechanism for Phase B. Assuming s_1 and s_2 reveal their true valuations, the *WDP* is as follows:

$$GMAP = \max_{q_1, q_2} (q_1 \mu_b(\eta_1) + q_2 \mu_b(\eta_2) + f_b^q(q_1, q_2) - q_1 \mu_1(\eta_1) - q_2 \mu_2(\eta_2)) = \max_{q_1, q_2} (f_b^q(q_1, q_2) + 50q_1 + 70q_2)$$

Since the bonus given by f_b^q for an even distribution is larger than 20, which is the gain from assigning the second unit to s_2 as well, the solution is $q_1 = q_2 = 1$, yielding a surplus of $30 + 50 + 70 = 150$. To calculate the price for each seller s_j we must find the surplus given that s_j does not participate, meaning that the other trader is allocated both units.

$$GMAP_{-1} = 70 \cdot 2 = 140, \quad GMAP_{-2} = 50 \cdot 2 = 100$$

This takes into account the loss of the bonus for quantity distribution. The profit of s_1 should be $150 - 140 = 10$, and the profit of s_2 should be 50. The final prices are therefore:

$$p(\eta_1) = 105 + 10 = 115, \quad p(\eta_2) = 75 + 50 = 125.$$

5.6 Two-Sided Multiattribute Call Market

Up to the end of this chapter I focus on *two-sided* multiattribute auctions, where multiple buyers and sellers submit bids, and the objective is to construct a set of deals maximizing overall surplus. The typical context I expect such an exchange to take place in is different than the typical setting I considered for procurement auctions. Rather than negotiating high-stake contracts for long-term supply of goods and services, an exchange is more likely to be used for commodities, such as cars or PCs. Therefore, some of the key preferences considered before, such as the distribution of quantity, do not play a significant role here.

Previous research on such auctions includes works by Fink et al. (2004) and Gong (2002), both of which consider a matching problem for *continuous* double auctions (CDAs), where deals are struck whenever a pair of compatible bids is identified.

In a *call market*, in contrast, bids accumulate until designated times (e.g., on a periodic or scheduled basis) at which the auction *clears* by determining a comprehensive match over the entire set of bids. Because the optimization is performed over an aggregated scope, call markets often enjoy liquidity and efficiency advantages over CDAs (Economides and Schwartz, 1995). In the interim between clears, call markets may also disseminate *price*

quotes providing summary information about the state of the auction (Wurman et al., 2001). Such price quotes are often computed based on hypothetical clears, and so the clearing algorithm may be invoked more frequently than actual market clearing operations. The advantages of price feedbacks in multiattribute call markets were studied by Lochner (2008).

Clearing a multiattribute CDA is much like clearing a one-sided multiattribute auction. Because nothing happens between bids, the problem is to match a given new bid (say, an offer to buy) with the existing bids on the other (sell) side. Multiattribute call markets are potentially much more complex. Constructing an optimal overall matching may require consideration of many different combinations of trades, among the various potential trading-partner pairings. The problem can be complicated by restrictions on overall assignments, as expressed in side constraints (Kalagnanam et al., 2001).

The goal of this part of the work is to explore tradeoffs between expressive power of agent bids and computational properties of auction clearing. I conduct the exploration independent of any consideration of strategic issues bearing on mechanism design. As with analogous studies of combinatorial auctions (Nisan, 2000), I intend that tradeoffs quantified in this work can be combined with incentive factors within a comprehensive overall approach to multiattribute auction design. I therefore treat the reported utility, through bids, as a faithful representation of the preferences.

From the starting point of Theorem 5.1, I address the two parts of the problem, *MMP* and *GMAP*. After considering *MMP*, I develop a family of network flow problems that capture corresponding classes of *GMAP* optimizations. Experimental trials provide preliminary confirmation that the network formulations provide useful structure for implementing clearing algorithms.

5.7 Utility Representation and *MMP*

I first consider the problem of finding a best deal between pairs of traders. *MMP* is an optimization problem over $\mu_b(\cdot) - \mu_s(\cdot)$, and therefore depends pivotally on the representation of $\mu(\cdot)$ in the bid. Note that issues of utility representation and *MMP* apply to a broad class of multiattribute mechanisms, beyond multiattribute call markets. For example, the complexity results contained in this section apply equally to the bidding problem faced by sellers in second-score auctions, given a published buyer $\mu_b(\cdot)$. As mentioned earlier, *MMP* comes up even in a price driven auction, such as the GAI auctions discussed earlier, when optimizing the difference between $\mu(\cdot)$ and current prices.

The simplest representation of $\mu(\cdot)$ is a direct enumeration of configurations and asso-

ciated quantities and payments. This approach treats the configurations as atomic entities, making no use of attribute structure. A common and inexpensive enhancement is to enable a trader to express *sets* of configurations, by specifying subsets of the domains of component attributes. Associating a single quantity and payment with a set of configurations expresses indifference among them; hence I refer to such a set as an *indifference range*.³ Indifference ranges include the case of attributes with a natural ordering, in which a bid specifies a minimum or maximum acceptable attribute level. The use of indifference ranges can be convenient for *MMP*. The compatibility of two indifference ranges is simply found by testing set intersection for each attribute, as demonstrated by the decision-tree algorithm of Fink et al. (2004).

Alternatively, bidders may specify their WTP $\mu(\cdot)$ in terms of compact functional forms. Enumeration based representations, even when enhanced with indifference ranges, are ultimately limited by the exponential size of attribute space. Functional forms may avoid this explosion, but only if $\mu(\cdot)$ reflects structure among the attributes. Moreover, even given a compact specification of $\mu(\cdot)$, we gain computational benefits only if we can perform the matching without expanding the $\mu(\cdot)$ values of an exponential number of configuration points.

Under fully additive utility representations of both sides, *MMP* is simple—the optimal match over Θ reduces to finding the optimal match separately for each attribute. A common scenario in practical procurement has the buyer define an additive scoring function, while suppliers submit enumerated offer points or indifference ranges. This model is still very amenable to *MMP*: for each element in a supplier’s enumerated set, we optimize each attribute by finding the point in the supplier’s allowable range that is most preferred by the buyer.

A special type of scoring (more particularly, cost) function was defined by Bichler and Kalagnanam (2005) and called a *configurable offer*. This idea is geared towards procurement auctions: assuming suppliers are usually comfortable with expressing their preferences in terms of cost that is quasi-linear in every attribute, they can specify a price for a base offer, and additional cost for every change in a specific attribute level. This model is essentially a “pricing out” approach. For this case, *MMP* can still be optimized on a per-attribute basis.

As advocated in previous chapters, we would often prefer the more flexible GAI representation. As GAI retains the summation structure, the complexity of the matching is similar to the complexity of optimizing a single function, since the sum function is in the

³These should not be mistaken with *indifference curves*, which express dependency between the attributes. Indifference curves can be expressed by the more elaborate utility representations discussed below.

form (6.4) as well, as discussed in Section 3.3.2.

5.8 Solving *GMAP* under Allocation Constraints

The problem of clearing a multiattribute call market is a combinatorial optimization problem, which is NP-hard with an input size that is twice exponential in the number of attributes. The *MMP-GMAP* decomposition is a significant simplification, but the space still needs to be severely restricted in order to allow reasonably tractable algorithms. I therefore define constraints that map to classes of bids, and inspect the complexity of optimization under various combinations of these restrictions.

The main restriction is called *configuration parity*. Configuration parity reduces the expressiveness of preferences in another dimension: instead of the quantity factor, traders can only specify \bar{q}_b and q_b . Traders express bounds on the total quantity they wish to obtain, and they cannot specify how that quantity is divided between suppliers. This is quite restrictive, but as argued above may be reasonably realistic for an exchange.

Definition 5.4. A configuration parity bid $\mathcal{B}_{\hat{b}}$ contains the following components:

- a representation of a single-unit utility function $\bar{\mu}_{\hat{b}}(\cdot)$.⁴
- quantities $\bar{q}_{\hat{b}}$ and $q_{\hat{b}}$.
- a boolean flag *AGG*.

$\mathcal{B}_{\hat{b}}$ with *AGG* = true indicates that \hat{b} is willing to engage in any allocation T , with $T_{\hat{b}} = \{(\theta, q, \hat{b}, s)\}$ such that $q_{\hat{b}} \leq \sum_{t \in T_{\hat{b}}} q_t \leq \bar{q}_{\hat{b}}$ and pay a total of $\sum_{t \in T_{\hat{b}}} q_t \bar{\mu}_{\hat{b}}(\theta_t, s_t)$. With *AGG* = false, $\mathcal{B}_{\hat{b}}$ indicates the same with the additional restriction $|T_{\hat{b}}| = 1$.

It is straightforward to show that any willingness-to-pay represented by a configuration aggregating bid is within the restrictions of Theorem 5.1. In other words, if for any $T_{\hat{b}}$ the amount expressed by $\mathcal{B}_{\hat{b}}$ is $u_{\hat{b}}(T_{\hat{b}})$, then \hat{b} exhibits configuration aggregation and linear trade quantity. We can therefore apply the *MMP-GMAP* decomposition to find the optimal allocation given a set of buy and sell bids.

In the rest of this section I consider the clearing problem for configuration parity bids. I first define several special cases of such bids.

Aggregating Bids $\mathcal{B}_{\hat{b}}$ for which *AGG* = true

Non Aggregating Bids $\mathcal{B}_{\hat{b}}$ for which *AGG* = false

All or None (AON) Bids $\mathcal{B}_{\hat{b}}$ for which $q_{\hat{b}} = \bar{q}_{\hat{b}}$.

⁴Usually some *bidding language* structures the expression of $\bar{\mu}(\cdot)$ in some compact way, but I do not get into this topic and assume that it contains an enumeration of points.

Divisible Bids $\mathcal{B}_{\hat{b}}$ for which $q_{\hat{b}} = 0$.

These correspond to the same classes of bids expressiveness defined in Engel et al. (2006), where they are introduced as a set of constraints over the space of all possible combination of trades.

5.8.1 Notation and Graphical Representation

The clearing algorithms are based on network flow formulations of the underlying optimization problem (Ahuja et al., 1993). The network model is based on a bipartite graph, in which nodes on the left side represent buyers, and nodes on the right represent sellers. Traders may submit multiple bids with an *OR* semantics, meaning they are willing to trade based on any subset of their bids. We can treat OR bids by the same traders as different traders without changing the result. Therefore, B and S refer to the *bids*, but I use the terms “bids” and “bidders” interchangeably.

I define two graph families, one for the case of non-aggregating bids (called *single-unit*), and the other for the case of aggregating bids (called *multi-unit*).⁵ For both types, a single directed arc is placed from a buyer $i \in B$ to a seller $j \in S$ if and only if $h(MMP(i, j))$ is positive, where $h(MMP(\cdot))$ refers to the *value* of the *MMP* solution. I denote by $T(i)$ the set of potential trading partners of trader i , i.e., the nodes connected to buyer or seller i in the bipartite graph.

In the single-unit case, I define the weight of an arc (i, j) as $w_{ij} = h(MMP(i, j))$. For the multi-unit case, the weights are defined similarly, and the quantity \bar{q}_i is associated with the node for bid i . I also use the notation q_{ij} for the mathematical formulations to denote partial fulfillment of q_i for t that solves $MMP(i, j)$.

5.8.2 Handling Indivisibility and Aggregation Constraints

Under the restrictions of Theorem 5.1 and configuration parity, and when the solution to *MMP* is given, *GMAP* exhibits strong similarity to the problem of clearing double auctions with assignment constraints (Kalagnanam et al., 2001). A match in the bipartite representation corresponds to a potential trade in which assignment constraints are satisfied. Network flow formulations have been shown to model this problem under the assumption of indivisibility and aggregation for all bids. The novelty in this part of the work is the use

⁵In the next section, I introduce a hybrid form of graph accommodating mixes of the two trader categories.

of generalized network flow formulations for more complex cases where aggregation and divisibility may be controlled by traders.

Initially I examine the simple case in which all bids are non-aggregating. Observe that the optimal allocation is simply the solution to the well known *weighted assignment problem* (Ahuja et al., 1993) on the single-unit bipartite graph described above. The set of matches that maximizes the total weight of arcs corresponds to the set of trades that maximizes total surplus. Note that any form of (in)divisibility can also be accommodated in this model via the constituent *MMP* subproblems.

The next formulation solves the case in which all traders are aggregating and divisible. This case can be represented using the following linear program, corresponding to the multi-unit graph described above:

$$\begin{aligned}
& \max \sum_{i \in B, j \in S} w_{ij} q_{ij} \\
& \text{s.t.} \sum_{i \in T(j)} q_{ij} \leq \bar{q}_j \quad j \in S \\
& \quad \sum_{j \in T(i)} q_{ij} \leq \bar{q}_i \quad i \in B \\
& \quad q_{ij} \geq 0 \quad j \in S, i \in B
\end{aligned}$$

Recall that the q_{ij} variables in the solution represent the number of units that buyer i procures from seller j . This formulation is known as the *network transportation problem with inequality constraints*, for which efficient algorithms are available (Ahuja et al., 1993). It is a well known property of the transportation problem (and flow problems on pure networks in general) that given integer input values, the optimal solution to the corresponding linear program is guaranteed to be integer as well. Figure 5.1 demonstrates the transformation of a set of bids to a transportation problem instance.

The problem becomes significantly harder when aggregation is given as an option to bidders, requiring various enhancements to the basic multi-unit bipartite graph described above. In general, I consider traders that are either aggregating or not, with either divisible or AON offers.

Initially I examine a special case, which at the same time demonstrates the hardness of the problem but still carries computational advantages. Let one side (e.g., buyers) be defined as restrictive (AON and non-aggregating), and the other side (sellers) as unrestrictive (divisible and aggregating). This problem can be represented using the following integer

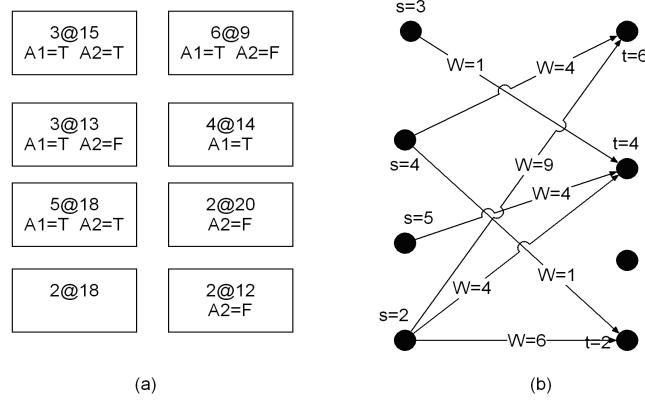


Figure 5.1: Multi-unit matching with two boolean attributes, A_1 and A_2 . (a) Bids, with offers to buy in the left column and offers to sell at right. $q@p$ indicates an offer to trade q units at price p per unit. (b) Corresponding multi-unit assignment model. W represents arc weights (unit surplus), s represents source (exogenous) flow, and t represents sink quantity.

programming formulation:

$$\begin{aligned}
 \max \quad & \sum_{i \in B, j \in S} w_{ij} q_{ij} \\
 \text{s.t.} \quad & \sum_{i \in T(j)} \bar{q}_i q_{ij} \leq \bar{q}_j \quad j \in S \\
 & \sum_{j \in T(i)} q_{ij} \leq 1 \quad i \in B \\
 & q_{ij} \in \{0, 1\} \quad j \in S, i \in B
 \end{aligned} \tag{5.6}$$

This formulation is a restriction of the *generalized assignment problem (GAP)* (Fisher et al., 1986). Although GAP is known to be NP-hard, it can be solved relatively efficiently by exact or approximate algorithms. GAP is more general than the formulation above as it allows buy-side quantities (\bar{q}_i above) to be different for each potential seller. This formulation is NP-hard as well (even the case of a single seller corresponds to the knapsack problem), illustrating the drastic increase in complexity when traders with different constraints are admitted to the same problem instance.

Other than the special case above, I found no advantage in limiting AON constraints when traders may specify aggregation constraints. Therefore, the next generalization allows any combination of the two boolean constraints, that is, traders choose among four bid types:

NI Bid AON and not aggregating.

AD Bid allows aggregation and divisibility.

AI Bid AON, allows aggregation (quantity can be aggregated across configurations, as long as it sums to the whole amount).

ND No aggregation, divisibility (one trade, but any quantity is acceptable).

In the next formulation B and S are each divided into four subsets corresponding to the types above. I indicate the set using the type name and a subscript, so that NI_b indicates buyers in NI . I omit the subscript when it is clear whether I refer to a buyer or to a seller.

I introduce an integer programming (IP) representation for the problem under these setting. The following additional variables are used in the formulation. Boolean (0/1) variables r_i and r'_j indicate whether buyer i and seller j participate in the solution (used for AON traders). Another indicator variable, y_{ij} , applied to non-aggregating buyer i and seller j , takes the value 1 iff i trades with j . For aggregating traders, y_{ij} is not constrained. The IP formulation is as follows.

$$\max \quad \sum_{i \in B, j \in S} W_{ij} q_{ij} \quad (5.7a)$$

$$\text{s.t.} \quad \sum_{j \in T(i)} q_{ij} = \bar{q}_i r_i \quad i \in AI_b \quad (5.7b)$$

$$\sum_{j \in T(i)} q_{ij} \leq \bar{q}_i r_i \quad i \in AD_b \quad (5.7c)$$

$$\sum_{i \in T(j)} q_{ij} = \bar{q}_j r'_j \quad j \in AI_s \quad (5.7d)$$

$$\sum_{i \in T(j)} q_{ij} \leq \bar{q}_j r'_j \quad j \in AD_s \quad (5.7e)$$

$$x_{ij} \leq \bar{q}_i y_{ij} \quad i \in ND_b, j \in T(i) \quad (5.7f)$$

$$x_{ij} \leq \bar{q}_j y_{ij} \quad j \in NI_s, i \in T(j) \quad (5.7g)$$

$$\sum_{j \in T(i)} y_{ij} \leq r_i \quad i \in NI_b \cup ND_b \quad (5.7h)$$

$$\sum_{i \in T(j)} y_{ij} \leq r'_j \quad j \in NI_s \cup ND_s \quad (5.7i)$$

$$\text{int } q_{ij} \quad (5.7j)$$

$$y_{ij}, r'_j, r_i \in \{0, 1\} \quad (5.7k)$$

Problem (5.7) has additional structure as a *generalized min-cost flow* problem with integral flow.⁶ A generalized flow network is a network in which each arc may have a *gain factor*, in addition to the pure network parameters (which are flow limits and costs). Flow in an arc is then multiplied by its gain factor, so that the flow that enters the end node of

⁶Constraint (5.7j) could be omitted (yielding computational savings) if non-integer quantities are allowed. Here and henceforth I assume the harder problem, where divisibility is with respect to integers.

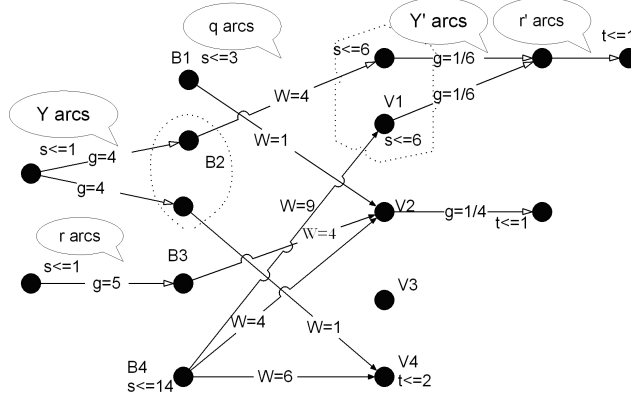


Figure 5.2: Generalized network flow model. $B1$ is a buyer in AD , $B2 \in NI$, $B3 \in AI$, $B4 \in ND$. $V1$ is a seller in ND , $V2 \in AI$, $V4 \in AD$. The g values represent arc gains.

an arc equals the flow that entered from its start node, multiplied by the gain factor of the arc. The network model can in turn be translated into an IP formulation that captures such structure.

The generalized min-cost flow problem is well-studied and has a multitude of efficient algorithms (Ahuja et al., 1993). The faster algorithms are polynomial in the number of arcs and the logarithm of the maximal gain, that is, performance is not strongly polynomial but is polynomial in the size of the input. The main benefit of this graphical formulation to the matching problem is that it provides a very efficient linear relaxation. Integer programming algorithms such as branch-and-bound use solutions to the linear relaxation instance to bound the optimal integer solution. Since network flow algorithms are much faster than arbitrary linear programs (generalized network flow simplex algorithms have been shown to run in practice only 2 or 3 times slower than pure network min-cost flow (Ahuja et al., 1993)), I expect a branch-and-bound solver for the matching problem to show improved performance when taking advantage of network flow modeling.

The network flow formulation is depicted in Figure 5.2. Non-restrictive traders are treated as in Figure 5.1. For a non-aggregating buyer, a single unit from the source will saturate up to one of the y_{ij} for all j , and be multiplied by \bar{q}_i . If $i \in ND$, the end node of y_{ij} will function as a sink that may drain up to \bar{q}_i of the entering flow. For $i \in NI$ an indicator (0/1) arc r_i is used, on which the flow is multiplied by \bar{q}_i . Trader i trades the full quantity iff $r_i = 1$.

At the seller side, the end node of a q_{ij} arc functions as a source for sellers $j \in ND$, in order to let the flow through y'_{ij} arcs be 0 or \bar{q}_j . The flow is then multiplied by $\frac{1}{\bar{q}_j}$ so 0/1 flows enter an end node which can drain either 1 or 0 units. For sellers $j \in NI$ arcs r'_j ensure AON similarly to arcs r_j for buyers.

Having established this framework, we are ready to accommodate more flexible versions of the constraints. The first generalization is to relax the boolean choice between AON and divisibility, allowing any minimal quantity q . In the network flow instance we simply need to turn the node of the constrained trader i (e.g., the node $B3$ in Figure 5.2) to a sink that can drain up to $\bar{q}_i - q_i$ units of flow. The integer program (5.7) can be also easily changed to accommodate this extension.

Using gains, we can also apply another popular type of side constraint, called *batch size*. A batch size constraint relaxes linear trade quantity, allowing trader to specify an integer β such that the total quantity traded is must equal $\gamma\beta$ for any integer γ . If a trader specifies a batch size β , we change the gain on the r arcs to β , and set the available flow of its origin to the maximal number of batches \bar{q}_i/β .

5.8.3 Nonlinear Value of Quantity

A key assumption in the results up to this point is linear trade quantity, which allows us to restrict attention to per-unit valuations. Divisibility without linear trade quantity allows expression of concave willingness-to-pay functions, corresponding to convex preference relations. Bidders may often wish to express non-convex offer sets, for example, due to fixed costs or switching costs in production settings (Schvartzman and Wellman, 2007).

The framework can be extended to accommodate nonlinear valuation of quantity in the form of enumerated payment *schedules*—that is, defining a function $\hat{\mu} : \Theta \times \bar{d}(q) \rightarrow \mathbb{R}^+$ ($\bar{D}(q) \subseteq D(q)$) which assigns values $\mu(x, q)$ for a select set of quantities q .

Definition 5.5. A schedule bid $\mathcal{B}_{\hat{b}}$ is a set of configuration parity bids with XOR semantics, that is \hat{b} is willing to engage in an allocation and payment implied by exactly one of the bids in the set.

To handle nonlinear valuations, the network is augmented to include flow possibilities corresponding to each of the XOR bids, plus additional structure to enforce exclusivity among them. In other words, the network treats the offer for a given quantity as in Section 5.8.2, and embeds this in an XOR relation to ensure that each trader picks only one allocation implied by one of the bids. Since for each such quantity choice we can apply Theorem 5.1, the solution we get is in fact the solution to *GMAP*.

The network representation of the XOR relation (which can be embedded into the network of Figure 5.2) is depicted in Figure 5.3. For simplicity I assume that all bids are divisible or AON, but minimal quantities can be treated as above. For a trader i with K XOR bids, I define dummy variables, z_i^k , $k = 1, \dots, K$. Since we consider trades between

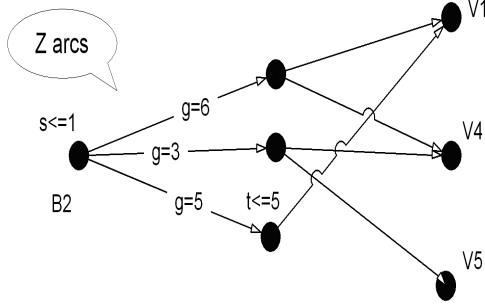


Figure 5.3: Extending the network flow model to express an XOR over quantities. Buyer B2 has 3 XOR points for 6, 3, or 5 units.

every pair of quantity points we also have q_{ij}^k , $k = 1, \dots, K$. For buyer $i \in AI$ with XOR points at quantities \bar{q}_{ik} , replace (5.7b) with the following constraints:

$$\begin{aligned}
 \sum_{j \in T(i)} q_{ij}^k &= \bar{q}_{ik} z_i^k \quad k = 1, \dots, K \\
 \sum_{k=1}^K z_i^k &= r_i \\
 z_i^k &\in \{0, 1\} \quad k = 1, \dots, K
 \end{aligned} \tag{5.8}$$

Figure 5.3 depicts the generalized network flow formulation.

5.8.4 Homogeneity Constraints

The model (5.7) handles constraints over the aggregation of quantities from different trading partners. When aggregation is allowed, the formulation permits trades involving arbitrary combinations of configurations. A homogeneity constraint (Bichler and Kalagnanam, 2005) restricts such combinations, by requiring that configurations aggregated in an overall deal must agree on some or all attributes.

In the presence of homogeneity constraints (as shown by Theorem 5.1) we can no longer apply the convenient separation of *GMAP* into *MMP* plus global bipartite optimization, as the solution to *GMAP* may include trades not part of any *MMP* solution. For example, let buyer b specify an offer for maximum quantity 10 of various acceptable configurations, with a homogeneity constraint over the attribute “color”. This means b is willing to aggregate deals over different trading partners and configurations, as long as all are the same color. If seller s can provide 5 blue units or 5 green units, and seller s' can provide only 5

green units, we may prefer that b and s trade on green units, even if the local surplus of a blue trade is greater.

Let $\{x_1, \dots, x_H\}$ be attributes that some trader constrains to be homogeneous. To preserve the network flow framework, we need to consider, for each trader, every point in the product domain of these homogeneous attributes. Thus, for every assignment \hat{x} to the homogeneous attributes, we compute $MMP(b, s)$ under the constraint that configurations are consistent with \hat{x} . This is the same approach as in Section 5.8.3: solve the global optimization, such that the alternative \hat{x} assignments for each trader are combined under XOR semantics, thus enforcing homogeneity constraints.

The size of this network is exponential in the number of homogeneous attributes, since we need a node for each point in the product domain of all the homogeneous attributes of each trader.⁷ Hence this solution method will only be tractable in applications where the traders can be limited to a small number of homogeneous attributes. It is important to note that the graph needs to include a node only for each point that potentially matches a point on the other side. It is therefore possible to make the problem tractable by limiting one of the sides to a less expressive bidding language, and by that limit the set of potential matches. For example, if sellers submit bounded sets of XOR points, we need to consider only the points in the combined set offered by the sellers, and the reduction to network flow is polynomial regardless of the number of homogeneous attributes.

If such simplifications do not apply, it may be preferable to solve the global problem directly as a single optimization problem. I provide the formulation for the special case of divisibility under configuration parity. Let i index buyers, j sellers, and H_i, H_j represent the sets of homogeneous attributes respectively. Variable $x_{ij}^h \in D(X_h)$ takes the value of attribute X_h in the trade between buyer b_i and seller s_j . Integer variable q_{ij} represents the quantity of the trade (zero for no trade) between b_i and s_j . The complexity is NP-hard as an Integer Programming formulation, and in addition it introduces potential non-linearity

⁷If traders differ on which attributes they express such constraints, we can limit consideration to the relevant alternatives. The complexity will still be exponential, but in the maximum number of homogeneous attributes for any pair of traders.

Aggregation	Hom. attr.	Min. quantity	linear tr. quant.	Technique	Complexity
No aggregation	N/A	Any	Not required	Assignment problem	Polynomial
All aggregate	None	Divisible	Required	Transpor. problem	Polynomial
One side	None	Aggr side div.	Aggr. side	GAP	NP-hard
Optional	None	Any, batch	Required	Generalized ntwrk flow	NP-hard
Optional	Bounded	Any, batch	Bounded size schdl.	Generalized ntwrk flow	NP-hard
Optional	Not bounded	Any, batch	Not required	Non-linear integer opt	NP-hard

Table 5.3: Mapping from combinations of allocation constraints to the solution methods of *GMAP*.

in the form of $\mu(\cdot)$.

$$\begin{aligned}
& \max \sum_{b_i \in B, s_j \in S} [q_{ij} \mu_{b_i}(x_{ij}, s_j) - q_{ij} \mu_{s_j}(x_{ij}, b_i)] \\
& \sum_{j \in S} q_{ij} \leq \bar{q}_i \quad i \in B \\
& \sum_{i \in B} q_{ij} \leq \bar{q}_j \quad j \in S \\
& x_{1_j}^h = x_{2_j}^h = \dots = x_{|B|_j}^h \quad s_j \in S, h \in \{1, \dots, H_j\} \\
& x_{i_1}^h = x_{i_2}^h = \dots = x_{i_{|S|}}^h \quad b_i \in B, h \in \{1, \dots, H_i\}
\end{aligned} \tag{5.9}$$

Table 5.3 summarizes the mapping I presented from allocation constraints to the complexity of solving *GMAP*. *One Side* means that one side aggregates and divisible, and the other side is restrictive. *Batch* means that traders may submit batch sizes. Configuration parity is assumed for all cases but the first. “Down to \bar{q} ” means divisibility with minimum quantity.

5.9 Experimental Results

I approach the experimental aspect of this work with two objectives. First, to obtain a general idea of the sizes and types of clearing problems that can be solved under given reasonable time constraints. Another goal is to compare the performance of a straightforward integer program as in (5.7) with an integer program that is based on the network formulations developed here. Since I used CPLEX, a commercial optimization tool, the second objective could be achieved to the extent that CPLEX can take advantage of network structure present in a model.

I found that in addition to the problem size (in terms of number of traders), the percentage of aggregating traders plays a crucial role in determining complexity. When most of

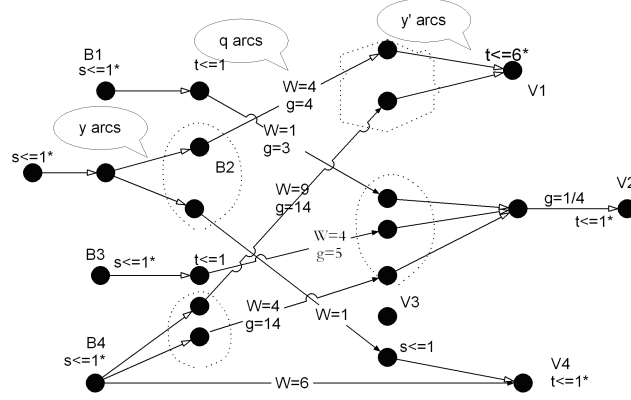


Figure 5.4: Alternative generalized network flow model. $B1$ is a buyer in AD , $B2 \in AI$, $B3 \in NI$, $B4 \in ND$. $V1$ is a seller in AD , $V2 \in AI$, $V4 \in ND$. The g values represent arc gains, and W values represent weights.

the traders are aggregating, problems of larger sizes can be solved quickly. For example, the IP model solved instances with 600 buyers and 500 sellers, where 90% of them are aggregating, in less than two minutes. When the aggregating ratio was reduced to 80% for the same data, solution time was just under five minutes.

These results were obtained because the network model (Figure 5.2) assigns a regular node to each aggregating trader, and treats non-aggregating traders as the special case. This seem to miss the fact that non-aggregation should be easier to handle (for the same reason the assignment problem is easier than the transportation problem). This motivated me to develop a new network model. Rather than treat non-aggregating traders as a special case, the new model takes advantage of the single-unit nature of non-aggregating trades, and treats the aggregating traders as the special case. This new model outperformed the other models on most problem instances, exceptions being those where aggregating traders constitute a vast majority (at least 80%).

This new model (Figure 5.4) has a single node for each non aggregating trader, with a single-unit arc designating a match to another non-aggregating trader. An aggregating trader has a node for each potential match, connected (via y arcs) to a mutual source node. Unlike the previous model I allow fractional flow for this case, representing the traded fraction of the buyer's total quantity. This fraction is then multiplied by the gain factor of the subsequent q arc, and the integrality constraint there ensures that the traded quantity remains integer.

I tested all three models on random data in the form of bipartite graphs encoding *MMP* solutions. In my experiments, each trader has a maximum quantity uniformly distributed over $[30, 70]$, and minimum quantity uniformly distributed from zero to the maximal quan-

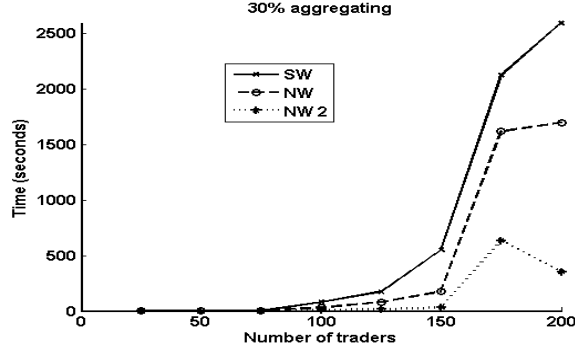


Figure 5.5: Average performance of models when 30% of traders aggregate.

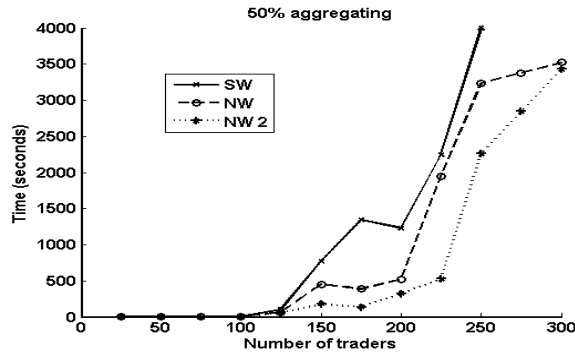


Figure 5.6: Average performance of models when 50% of traders aggregate.

tity. For each pair of a buyer and a seller, the solution to *MMP* is drawn in two steps: first, it is chosen to be positive (meaning that the buyer and seller *match*) with probability 0.75. For each matching pair the surplus is drawn uniformly from $[10, 70]$.

The size of the problem is defined by the number of traders on each side, but the problem complexity depends in fact on the product $|B||S|$. In all of the tests shown, for a given number of traders N , I define $|B|$ and $|S|$ according to the worst case $|B| = |S| = \frac{N}{2}$. Each data point is averaged over six samples. In figures 5.5–5.8, the direct IP (Eq. 5.7) is designated “SW”, the first network model (Figure 5.2) “NW”, and the revised network model (Figure 5.4) “NW 2”.

Figures 5.5–5.7 depict the comparison of the three models as a function of the number of traders, given various percentages of aggregating traders. Figure 5.8 shows how the various models are affected by a change in the percentage of aggregating traders, holding the problem size fixed.⁸

⁸All tests were performed on Intel 3.4 GHz processors with 2048 KB cache. Tests that did not complete by the one-hour time limit were recorded as 4000 seconds.

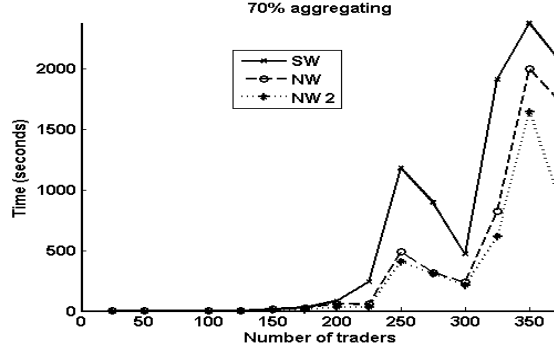


Figure 5.7: Average performance of models when 70% of traders aggregate.

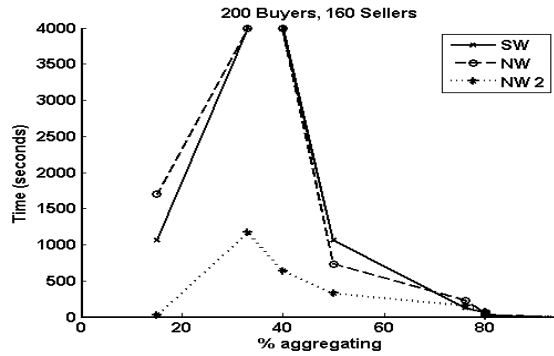


Figure 5.8: Performance of models when varying percentage of aggregating traders

Due to the integrality constraints, I could not test available algorithms specialized for network-flow problems on the test problems. Thus, I cannot fully evaluate the potential gain attributable to the network structure. However, the model I built based on the insight from the network structure clearly provided a significant speedup, even without using a special-purpose algorithm. Model *NW 2* provided speedups by a factor of 4–10 over the model *SW*. This was consistent for all the problem sizes I tested, including the smaller sizes for which the speedup is not visually apparent on the chart.

5.10 Summary

I present a preference handling approach for the domain of multi-unit multiattribute auctions. I show how GAI decomposition can be leveraged to express structured MUMA preferences, and that its functional form naturally expresses non-regularities in traders' utilities. Specific restrictions allow the decoupling of the multi-unit problem from the

multiattribute domain, leading to an auction mechanism that yields approximate efficiency under approximate incentive properties, and takes advantage of GAI structure in both the multiattribute and the MUMA domains.

Furthermore, I study the clearing problem of multiattribute exchanges. The space of feasible such mechanisms is constrained by computational limitations imposed by the clearing process. The extent to which the space of feasible mechanisms may be quantified a priori will facilitate the search for such exchanges in the full mechanism design problem.

Specifically, I investigate the space of two-sided multiattribute auctions, focusing on the relationship between constraints on the offers traders can express through bids, and the resulting computational problem of determining an optimal set of trades. Based again on the decoupling result of the multiattribute matching problem from the global allocation problem, I developed network flow models for the overall clearing task, which facilitate classification of problem versions by computational complexity, and provide guidance for developing solution algorithms and relaxing bidding constraints.

Chapter 6

CUI Networks

6.1 Motivation

The motivation for the development of graphical models for multiattribute utility functions is described in Chapter 1. Foremost, a graphical model usually supports a compact representation of lower dimension, and reduces the amount of data needed to be elicited and maintained. Furthermore, the graphical representation encapsulates the dependency structure between the attributes. This qualitative preference information can potentially be reused over time, while the actual utility numbers may be changing. For example, it is reasonable that a particular CDI map can be used for repeated instances of a GAI auction. Finally, the graphical structure may help in the design of efficient reasoning algorithms (e.g., the GAI network facilitates optimization algorithms). As described in the previous chapters, graphical models suggested so far rely on notions of additivity, and in particular on GAI (Boutilier et al., 2001; Gonzales and Perny, 2004). The most obvious benefit of a model based instead on (conditional) utility independence is the generality admitted by a weaker independence condition, in comparison to the additivity required by CDI, CAI, and GAI (see discussion in Section 3.1). Perhaps for this reason, concluding their pioneer work on graphical models for conditionally additive independent preferences, Bacchus and Grove (1995) suggest investigating graphical models of other independence concepts, in particular utility independence.

In Chapter 4 I introduced auctions that employ the GAI and CDI scheme. This scheme possesses certain advantages for auction design. First, as an additive form, a GAI price structure can naturally represent decomposition of prices. Second, the GAI form becomes a fully additive decomposition in the extreme case, in which all GAI elements are of size one. This can help the adaptation of the mechanism for a market in which the current practice is limited to the fully additive form.

Sometimes, however, additive forms may not be sufficient to achieve a low-dimensional decomposition. In Chapter 4, I introduced an example of a CDI map and GAI network con-

structed for a hard drive procurement problem. In practice, the problem will have a larger number of attributes, describing a finer grain characterization of hard drives. For example, performance can be described, in addition to the attributes Transfer Rate and RPM, by the attributes Latency and Access Time. It is plausible that all of these attributes are complements, because one performance measurement is utilized better when the value of another is higher. In addition, they may all be complements of Volume, because a larger volume requires better performance. This results in a clique of size 5. Elicitation of utility functions of this dimension already poses a much tougher challenge. As mentioned in Chapter 3, utility independence can accommodate complements, and therefore we can expect to achieve lower dimensionality with an appropriate graphical modeling. For the reasons mentioned above, I expected that GAI will still be preferable for iterative auction design, but the new graphical modeling can be applied for other forms of trading, in particular for direct revelation mechanisms. It is also important to note that the model presented in this chapter is not by any means restricted to the trading domain. This model applies to any cardinal utility function, in particular to the vNM function for decision problems under uncertainty.

The utility tree (Von Stengel, 1988) is the only previous graphical decomposition suggested for utility functions that employs the original, non-symmetric notion of utility independence. The utility trees however lead to a *hierarchical* decomposition. In contrast, the approach taken in this chapter is closer to modern graphical models, in which *conditional* independence assumptions lead to decomposition to independent functions. This opens the way to adaptation of reasoning tools as the optimization algorithms described later. I therefore do not directly compare CUI networks with utility trees.

6.1.1 CUI Compared to other Independence Concepts

Bacchus and Grove (1995) exemplify the difference between additive and utility independence on a simple state space of two boolean attributes: Health and Wealth. In their example, shown in Table 6.1, the attributes are not additive independent (as can be immediately seen using preference differences), because H and W are complements: $u(H, W) > u(H, \neg W) + u(\neg, W)$. This pair of attributes would be substitutes if, for example, $u(W, H) = 4$ and $u(W, \neg H) = 3$. In both cases H and W are nonetheless preferential independent, since we always prefer to be richer (all else being equal) and we always prefer to be healthier (all else being equal). For boolean variables, first-order preferential independence and *first-order utility independence* are equivalent (we always prefer lotteries that assign higher probability to the more preferred level) and therefore Health and Wealth are also UI of each other.

	H	$\neg H$
W	5	1
$\neg W$	2	0

Table 6.1: Utility values for the Health and Wealth example (Bacchus and Grove, 1995).

The next example shows the following two phenomena:

1. A case where PI holds but not UI.
2. No cardinal independence holds, except for a non-symmetric CUI.

The first phenomenon requires the domains of H , or W , or both to include at least three values.¹ In order to demonstrate conditional independence, a third attribute must be added to the outcome space: location (L), indicating whether we live in the city or in the countryside (Table 6.2). W_r means rich, W_m is medium income, W_p means poor. H_f is healthy and at top fitness, H_g means good health, and H_s means sick. L_{ci} stands for city location, and L_{co} means countryside location.

I begin by refuting the potential unconditional UI relations. In order to show that $UI(H, \{W, L\})$ does not hold, it is enough to find that it is violated for one pair of lotteries. Given the partial outcome (W_r, L_{ci}) we prefer the equal chance lottery over $\langle H_f, H_s \rangle$, whose expected utility is $\frac{12+5}{2}$, to the sure outcome H_g (value 8), whereas given (W_p, L_{ci}) we are indifferent (expected utility of 2 to both lotteries). Intuitively, it may be the case that the additional value we get from fitness (over good health) is higher if we are also rich, making it more significant than the value H_g adds over H_s . Similarly, $UI(W, \{H, L\})$ does not hold, by comparing the even-chance gamble over $\langle W_r, W_p \rangle$ and the sure outcome W_m , first given (H_f, L_{ci}) and then given (H_s, L_{ci}) .

Therefore, neither W nor H are utility independent, but it is easy to see that each is (first-order) preferential independent. L , however, is not: when we are rich we would rather live in the city, and the other way round when we are poor, except for the case of being poor and sick under which we prefer the city.

	L_{ci}			L_{co}		
	H_f	H_g	H_s	H_f	H_g	H_s
W_r	12	8	5	10	6	4
W_m	6	4	3	6	3	2
W_p	3	2	1	4	1.5	0

Table 6.2: Utility values for the Health, Wealth, and Location example.

Therefore, no symmetric independence condition exists here, and that rules out any additive or multiplicative independence, conditional or not, between any subsets of attributes.

¹I extend the domains of both W and H because in both PI intuitively holds, and I do not want UI to hold.

Also, since no single variable is unconditionally UI, no subset can be unconditionally UI. Further, the fact that preferences over L depend on the combination of H and W rules out a GAI decomposition of the form $\{W, L\}, \{W, H\}, \{H, L\}$.

We can, however, achieve decomposition using CUI. It is the case that $\text{CUI}(W, L|H)$, since each column on the left matrix (L_{ci}) is an affine transformation of its counterpart on the right side (L_{co}). For example, to transform the first column (H_f), multiply by $\frac{2}{3}$ and add 2.

This example illustrates the subtlety of utility independence. In particular, whereas preferences over L depend on W , W may still be (conditionally) UI of L . A CAI assumption for the same attributes must inevitably ignore the reversal of preferences over L for different values of W , hence a decision maker that will be queried for preferences under this assumption may not be able to provide meaningful answers.

The interaction with a system that requires preference representation normally requires the identification of structure, and then the extraction of the utility values that are required by the compact representation. It is therefore most important that these two tasks are as simplified as possible, whereas the functional form handled by the system may be more sophisticated. This is exactly the tradeoff made by CUI networks. It improves upon a GAI-based representation on these two most difficult tasks, for the following reason: a GAI condition can either be based on a collection of CAI conditions, or not. If it is (as is the case in Chapter 4), CUI networks achieve lower dimensionality (Section 6.6), and therefore easier elicitation. If it is not, the GAI condition is hard to identify, as discussed in Section 2.2. In CUI networks achieves lower dimension using an easier to verify independence condition, paying with a more complex functional form and algorithms, but these are handled behind the scenes.

6.1.2 Graphical Models for CUI

Founding a graphical model on UI is more difficult, in comparison for example to the additive conditions, as utility independence does not decompose the utility function as effectively as does additive independence. In particular, the condition $\text{UI}(Y, X)$ ensures that Y has a subutility function, but since X does not have one it is harder to carry on the decomposition into X . Hence in the case that X is large the dimensionality of the representation may remain too high. My approach therefore employs CUI conditions on large subsets Y , in which case the decomposition can be driven further by decomposing the conditional utility function of Y using more CUI conditions.

In the sequel I show how serial application of CUI leads to functional decomposition.

The corresponding graphical model, a CUI network, provides a lower-dimension representation of the utility function in which the preference information at each node depends only on the node and its parents. I demonstrate the use of CUI networks by constructing an example for a relatively complex domain. Next I elaborate on the technical and semantic properties of the model and knowledge required to construct it. Subsequent technical sections present optimization algorithms and techniques for further reducing the complexity of the representation.

6.2 CUI Networks

I begin by constructing a directed acyclic graph (DAG) representing a set of CUI conditions, followed by a derivation of the functional decomposition over the nodes of the DAG.

6.2.1 CUI DAG

Suppose that we obtain a set σ of CUI conditions on the variable set $A = \{x_1, \dots, x_n\}$, such that for each $x \in A$, σ contains a condition of the form

$$\text{CUI}(A \setminus (\{x\} \cup P(x)), x \mid P(x)).$$

In other words, the set $P(x)$ separates the rest of the variables from x . Such a $P(x)$ always exists, because for $P(x) = A \setminus \{x\}$ the condition above trivially holds. The set σ can be represented graphically as a directed acyclic graph by Procedure CUI-DAG.

Procedure:CUI-DAG

Choose an order over the set A (here assume ordering x_1, \dots, x_n for convenience);

Define the set of parents of x_1 as $Pa(x_1) = P(x_1)$;

for $i = 2, \dots, n$ **do**

Define $\tilde{Dn}(x_i)$ as the smallest subset of $\{x_1, \dots, x_{i-1}\}$ that satisfies the following condition:

$$[x_i \in Pa(x_j) \vee \exists k \in \{1, \dots, i-1\}. x_k \in \tilde{Dn}(x_i) \cap Pa(x_j)] \Rightarrow x_j \in \tilde{Dn}(x_i). \quad (6.1)$$

Define the parents of x_i to be the nodes in $P(x_i)$ which are not already descendants of x_i ,

$$Pa(x_i) = P(x_i) \setminus \tilde{Dn}(x_i).$$

We choose an order of the attributes, and iterate over the attributes in that order. For

each attribute x_i we set its set of parents, as the attributes in $P(x_i)$ that were not already defined to be its descendants. The formal definition of $\tilde{Dn}(x_i)$ in the procedure simply means the set of intermediate descendants of x_i : the set of nodes in x_1, \dots, x_{i-1} , for which (at this iteration) x_i is a parent or another descendant of x_i is a parent. Let $Dn(x)$ denote the final set of descendants of x . It is the set defined by Equation (6.1), when replacing $\{1, \dots, i-1\}$ with $\{1, \dots, n\}$. By their definitions, $Dn(x) \supseteq \tilde{Dn}(x)$, hence

$$Pa(x) \cup Dn(x) \supseteq Pa(x) \cup \tilde{Dn}(x) = P(x). \quad (6.2)$$

Proposition 6.1. *Consider the DAG defined by Procedure CUI-DAG for a set of attributes A . For any $x \in A$,*

$$CUI(A \setminus (\{x\} \cup Pa(x) \cup Dn(x)), x \mid Pa(x) \cup Dn(x)). \quad (6.3)$$

As an example, I show the construction of the structure on a small set of variables $A = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, for which we are given the following set of CUI conditions:

$$\begin{aligned} \sigma = \{ & CUI(\{x_4, x_5, x_6\}, x_1 \mid \{x_2, x_3\}), CUI(\{x_4, x_3, x_6\}, x_2 \mid \{x_1, x_5\}), \\ & CUI(\{x_2, x_4, x_6\}, x_3 \mid \{x_1, x_5\}), CUI(\{x_1, x_3, x_5\}, x_4 \mid \{x_2, x_6\}), \\ & CUI(x_6, x_5 \mid \{x_1, x_2, x_3, x_4\}), CUI(\{x_1, x_2, x_3, x_5\}, x_6 \mid x_4) \}. \end{aligned}$$

Construction of the network using the order implied by the indices results in the CUI DAG illustrated in Figure 6.1. The minimal separating set for x_1 is $\{x_2, x_3\}$. For x_2 , we get $\tilde{Dn}(x_2) = \{x_1\}$, and the only non-descendant variable that is required to separate it from the rest is x_5 , which is therefore its only parent. The rest of the graph is constructed in a similar way. When x_4 is placed, we find that $P(x_4) = \{x_2, x_6\}$. Therefore, x_4 becomes descendant of x_2 after x_2 is placed, in other words $Dn(x_2) = \tilde{Dn}(x_2) \cup \{x_4\} = \{x_1, x_4\}$.

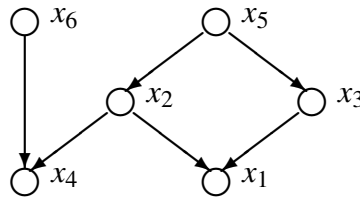


Figure 6.1: CUI DAG given σ and order x_1, \dots, x_6 .

Definition 6.1. *Let $u(A)$ be a utility function representing cardinal preferences \preceq^* over Θ . A CUI DAG for $u(\cdot)$ is a DAG, such that for any $x \in A$, (6.3) holds.*

Procedure CUI-DAG yields a CUI DAG by Proposition 6.1. For the other direction, any given CUI DAG G (in which parents and descendants are denoted by $Pa_G(\cdot), Dn_G(\cdot)$, respectively) can be constructed using Procedure CUI-DAG, as follows. Define $P(x) = Pa_G(x) \cup Dn_G(x)$ and a variable ordering according to the reverse topological order of G , and complete the execution of CUI-DAG. It is straightforward to show that the set of parents selected for each x_i is exactly $Pa_G(x_i)$, hence the result is a DAG that is identical to G .

6.2.2 CUI Decomposition

I now show how the CUI conditions, guaranteed by Proposition 6.1, can be applied iteratively to decompose $u(\cdot)$ to lower dimensional functions. The first step is to pick a variable ordering that agrees with the reverse topological order of the CUI DAG. To simplify the presentation, I rename the variables so the ordering is again x_1, \dots, x_n . The CUI condition (6.3) on x_1 implies the following decomposition, according to (2.2):

$$u(A) = f_1(x_1, Pa(x_1), Dn(x_1)) + g_1(x_1, Pa(x_1), Dn(x_1))u([A \setminus \{x_1\}]). \quad (6.4)$$

Note that $Dn(x_1) = \emptyset$.

Recall that $u([A \setminus \{x_1\}])$ on the right-hand side is the conditional utility function on A given that x_1 is fixed at an arbitrarily chosen reference point x_1^0 .

By applying the decomposition based on the CUI condition of x_2 on the conditional utility function $u([A \setminus \{x_1\}])$, we get

$$u([A \setminus \{x_1\}]) = f_2(x, Pa(x_2), Dn(x_2)) + g_2(x_2, Pa(x_2), Dn(x_2))u([A \setminus \{x_1, x_2\}]). \quad (6.5)$$

Note that $Dn(x_2) \subseteq \{x_1\}$, and x_1 is fixed to x_1^0 , hence f_2 and g_2 effectively depend only on x_2 and $Pa(x_2)$. This point is exploited below.

Substituting $u([A \setminus \{x_1\}])$ in (6.4) according to (6.5) yields:

$$u(A) = f_1 + g_1(f_2 + g_2u([A \setminus \{x_1, x_2\}])) = f_1 + g_1f_2 + g_1g_2u([A \setminus \{x_1, x_2\}]).$$

The arguments to the functions f_j, g_j are always $(x_j, Pa(x_j), Dn(x_j))$, hence I omit them for readability.

Continuing in this fashion we get

$$u(A) = \sum_{k=1}^{i-1} (f_k \prod_{j=1}^{k-1} g_j) + \prod_{j=1}^i g_j u([x_i, \dots, x_n]),$$

and apply the CUI condition of x_i ,

$$u([x_i, x_{i+1}, \dots, x_n]) = f_i(x_i, Pa(x_i), Dn(x_i)) + g_i(x_i, Pa(x_i), Dn(x_i))u([x_{i+1}, \dots, x_n]). \quad (6.6)$$

For convenience, I define the constant function $f_{n+1} \equiv u(x_1^0, \dots, x_n^0)$. Ultimately we obtain

$$u(A) = \sum_{i=1}^{n+1} [f_i(x_i, Pa(x_i), Dn(x_i)) \prod_{j=1}^{i-1} g_j(x_j, Pa(x_j), Dn(x_j))]. \quad (6.7)$$

The variable ordering is restricted to agree with the reverse topological order of the graph, hence in (6.6), $Dn(x_i) \subseteq \{x_1, \dots, x_{i-1}\}$. Therefore, all the variables in $Dn(x_i)$ on the right-hand side of (6.6) are fixed on their reference points, so f_i and g_i depend only on x_i and $Pa(x_i)$. Formally, let y_1, \dots, y_k be the variables in $Dn(x_i)$. With some abuse of notation, I redefine $f_i(\cdot)$ and $g_i(\cdot)$ as follows:

$$\begin{aligned} f_i(x_i, Pa(x_i)) &= f_i(x_i, Pa(x_i), y_1^0, \dots, y_k^0), \\ g_i(x_i, Pa(x_i)) &= g_i(x_i, Pa(x_i), y_1^0, \dots, y_k^0). \end{aligned} \quad (6.8)$$

Now (6.7) becomes

$$u(A) = \sum_{i=1}^{n+1} (f_i(x_i, Pa(x_i)) \prod_{j=1}^{i-1} g_j(x_j, Pa(x_j))). \quad (6.9)$$

This term is a decomposition of the multiattribute utility function to lower-dimensional functions, whose dimensions depend on the number of variables of $Pa(x)$. As a result, the dimensionality of the representation is reduced (as in Bayesian networks) to the maximal number of parents of a node plus one.

I illustrate how the utility function is decomposed in the example of Figure 6.1. We pick the ordering $x_4, x_1, x_6, x_3, x_2, x_5$ that agrees with the reverse topological order of the

graph (note that variables are not renamed here).

$$\begin{aligned}
u(A) &= f_4(x_4x_2x_6) + g_4(x_4x_2x_6)u([A \setminus \{x_4\}]) \\
u([A \setminus \{x_4\}]) &= f_1(x_1x_2x_3) + g_1(x_1x_2x_3)u([A \setminus \{x_4x_1\}]) \\
u([A \setminus \{x_4x_1\}]) &= f_6(x_6) + g_6(x_6)u([x_2x_3x_5]) \\
u([x_2x_3x_5]) &= f_3(x_3x_5) + g_3(x_3x_5)u([x_2x_5]) \\
u([x_2x_5]) &= f_2(x_2x_5) + g_2(x_2x_5)u([x_5]) \\
u([x_5]) &= f_5(x_5) + g_5(x_5)u(x_1^0, \dots, x_6^0)
\end{aligned}$$

Note that each f_i and g_i depends on x_i and its parents. Merging the above equations, and using the definition $f_7 \equiv u(x_1^0, \dots, x_6^0)$ produces

$$u(A) = f_4 + g_4f_1 + g_4g_1f_6 + g_4g_1g_6f_3 + g_4g_1g_6g_3f_2 + g_4g_1g_6g_3g_2f_5 + g_4g_1g_6g_3g_2g_5f_7. \quad (6.10)$$

Equation (6.9) establishes that $u(A)$ can be represented using a set of functions \mathcal{F} , that includes, for any $x \in A$, the functions (f_x, g_x) resulting from the decomposition (2.2) based on the CUI condition (6.3). This means that to fully specify $u(A)$ it is sufficient to obtain the data for functions in \mathcal{F} (this task is discussed in Section 6.4).

Definition 6.2. Let $u(A)$ be a utility function representing cardinal preferences \preceq^* over Θ . A CUI network for $u(\cdot)$ is a triplet (G, \mathcal{F}, A^0) . $G = (A, E)$ is a CUI DAG for $u(A)$, A^0 is a reference point, and \mathcal{F} is the set of functions $\{f_i(x_i, Pa(x_i)), g_i(x_i, Pa(x_i)) \mid i = 1, \dots, n\}$ defined above.

The utility value for any assignment to A can be calculated from the CUI network according to (6.9), using any variable ordering that agrees with the reverse topological order of the DAG. In this example, we can choose a different variable ordering than the one used above, such as $x_1, x_3, x_4, x_2, x_5, x_6$, leading to the following expression.

$$u(A) = f_1 + g_1f_3 + g_1g_3f_4 + g_1g_3g_4f_2 + g_1g_3g_4g_2f_5 + g_1g_3g_4g_2g_5f_6 + g_1g_3g_4g_2g_5g_6f_7.$$

This sum of products is different than the one in (6.10). However, it is based on the same CUI decompositions and therefore the same functions f_i and g_i .

6.2.3 Properties of CUI Networks

Based on Procedure CUI-DAG and the decomposition following it, I conclude the following.

Proposition 6.2. *Let A be a set of attributes, and σ a set of CUI conditions on A . If σ includes a condition of the form $CUI(A \setminus (x \cup Z_x), x \mid Z_x)$ for each $x \in A$, then σ can be represented by a CUI network whose dimensionality does not exceed $\max_x(|Z_x| + 1)$.*

Note that Z_x denotes here a minimal set of attributes (variables) that renders the rest CUI of x . This bound on the dimensionality will be obtained regardless of the variable ordering. We can expect the maximal dimension to be lower if the network is constructed using a good variable ordering. A good heuristic in determining the ordering would be to use attributes with smaller dependent sets first, so that the attributes with more dependents would have some of them as descendants. Based on such an ordering I expect the less important attributes to be lower in the topology, while the more crucial attributes would either be present higher or have a larger number of parents.

From this point I usually omit the third argument when referring to a CUI condition, as in $CUI(X, Y)$, which is taken equivalent to $CUI(X, Y \mid A \setminus (X \cup Y))$.

In order to achieve a low dimensional CUI network, we are required to detect CUI conditions over large sets. This may be a difficult task, and I address it through an example in Section 6.3. The task is made somewhat easier by the fact that the set has to be CUI of a single variable; note that the condition $CUI(Y, x)$ is weaker than the condition $CUI(Y, X)$ when $x \in X$. Furthermore, Section 6.6 shows how the dimensionality can be reduced if the initial CUI decomposition is not sufficiently effective.

Based on properties of CUI, we can read additional independence conditions off the graph. First, we observe that CUI has a composition property at the second argument.

Lemma 6.3. *Let $CUI(X, Y) [X, Y \subset A]$, and $CUI(B, C) [B, C \subset A]$. Then*

$$CUI(B \cap X, Y \cup C).$$

This property leads to the following claim, which allows us to derive additional CUI conditions once the graph is constructed.

Proposition 6.4. *Consider a CUI network for a set of attributes A . Define $Pa(X) = \bigcup_{x \in X} Pa(x)$ and $Dn(X) = \bigcup_{x \in X} Dn(x)$. Then for any $X \subset A$,*

$$CUI(A \setminus (X \cup Pa(X) \cup Dn(X)), X).$$

The proof is by recursion on X , using Lemma 6.3 and Proposition 6.1.

The next result presents the reverse type of independence: it defines a set of nodes that renders a set CUI of the rest. This dual perspective becomes particularly useful for optimization (Section 6.5), because optimization based on the preference order over an attribute is meaningful only when holding enough other attributes fixed to make it CPI or CUI of the rest. Let $Ch(X)$ denote the union of children of nodes in X , and let $An(X)$ denote all of the ancestors of nodes in X , in both cases excluding nodes which are themselves in X .

Proposition 6.5. *Consider a CUI network for a set of attributes A . $CUI(X, A \setminus (X \cup An(X) \cup Ch(X)))$ for any $X \subseteq A$.*

Finally, the following result relates CUI DAGs to CAI maps.

Proposition 6.6. *Let $G = (X, E)$ be a CAI map, and x_1, \dots, x_n an ordering over the nodes in X . Let $G' = (X, E')$ be the DAG such that there is a directed arc (x_i, x_j) in E' iff $i < j$ and $(x_i, x_j) \in E$. Then G' is a CUI DAG.*

Note, however, that CAI maps decompose the utility function over the maximal cliques, whereas CUI networks decompose over nodes and their parents. Section 6.6 bridges this gap. In addition, this result is used in Section 6.5.3.

6.3 CUI Modeling Example

To demonstrate the potential representational advantage of CUI networks we require a domain that is difficult to simplify otherwise. The example I use is the choice of a software package by an enterprise that wishes to automate its sourcing (strategic procurement) process. I focus on the software's facilities for running auction or RFQ (request for quotes) events, and tools to select winning suppliers either manually or automatically.

I identified nine key features of these kinds of software packages. In this choice scenario, the buyer evaluates each package on these nine features, graded on a discrete scale (e.g., one to five). The features are, in brief:

Interactive Negotiations (IN) allows a separate bargaining procedure with each supplier.

Multi-Stage (MS) allows a procurement event to be comprised of separate stages of different types.

Cost Formula (CF) buyers can formulate their total cost of doing business with each supplier.

Supplier Tracking (ST) allows long-term tracking of supplier performance.

MultiAttribute (MA) bidding over multiattribute items, potentially using a scoring function.²

Event Monitoring (EM) provides an interface to running events and real-time graphical views.

Bundle Bidding (BB) bidding for bundles of goods.

Grid Bidding (GB) adds a bidding dimension corresponding to an aspect such as time or region.

Decision Support (DS) tools for optimization and for aiding in the choice of the best supplier(s).

We observe first that additive independence does not widely apply in this domain. For example, Multi-Stage makes several other features more useful or important: Interactive Negotiations (often useful as a last stage), Decision Support (to choose which suppliers proceed to the next stage), and Event Monitoring (helps keep track of how useful was each stage in reducing costs). Conversely, in some circumstances Multi-Stage can substitute for the functionality of other features: MultiAttribute (by bidding on different attributes in different stages), Bundle Bidding (bidding on separate items in different stages), Grid Bidding (bidding on different time/regions in different stages) and Supplier Tracking (by extracting supplier information in a “Request for Information” stage). The potential dependencies for each attribute are shown in Table 6.3.

Attr	Complements	Substitutes	CUI set
EM	CF ST MS		IN,DS,MA,GB,BB
IN	ST MS MA		EM,CF,ST,DS,GB,BB
CF	EM MS DS MA GB BB	DS	MA,GB,BB
ST	EM MS IN DS	MA	IN,CF,GB,BB
MA	IN DS CF	MS BB ST GB	GB,BB
MS	DS EM IN ST	GB BB MA CF	MA,GB,BB
DS	CF MA GB ST BB MS		IN,EM
GB	CF DS	MA MS BB	MA,BB
BB	CF DS	MA MS GB	MA,GB

Table 6.3: Dependent and independent sets for each attribute.

²I hope the fact that the software itself may include facilities for multiattribute decision making does not cause undue confusion. Naturally, I consider this an important feature.

The presence of a complement or substitute relation precludes additive independence. From this fact we can identify a set of six attributes that must be mutually (additive) dependent: $\{BB, GB, DS, MA, MS, CF\}$. In consequence, the best-case dimensionality achieved by a CAI map (and other CAI-based representations, see Chapter 2), for this domain would be six, the size of the largest maximal clique.

In order to construct a CUI network we first identify, for each attribute x , a set Y that is CUI of it. We can first guess such a set according to the complement/substitute information in Table 6.3; typically, the set of attributes that are neither complements nor substitutes would be CUI. This is the approach taken for the attributes EM and DS . However, attributes that are complements or substitutes may still be CUI of each other, and we therefore attempt to detect and verify potentially larger CUI sets. Keeney and Raiffa (1976) provide several useful results that can help in detection of UI, and those results can be generalized to CUI. In particular they show that we can first detect a conditional preferential independence condition in which one element is also CUI. Based on this result, in order to verify for example that

$$\text{CUI}(\{BB, GB, MA\}, CF \mid A \setminus \{BB, GB, MA, CF\}), \quad (6.11)$$

the following two conditions are sufficient:

$$\text{CPI}(\{BB, GB, MA\}, CF \mid A \setminus \{BB, GB, MA, CF\}), \quad (6.12)$$

$$\text{CUI}(BB, \{GB, MA, CF\} \mid A \setminus \{BB, GB, MA, CF\}). \quad (6.13)$$

Detection and verification of these conditions are also discussed by Keeney and Raiffa. For the example, we observe that the features BB , GB , and MA each add a qualitative element to the bidding. Each bidding element is best exploited when cost formulation is available, so complements CF . The complementarity is similar for each feature, thus implying (6.12). Moreover, BB is a crucial feature and therefore the risk attitude towards it is not expected to vary with the level of CF , MA , and GB , and that implies (6.13), together leading to (6.11).

In a similar fashion, we observe that the nature of the substitutivity of the three mechanisms BB, GB, MA in MS is similar: each can be simulated using multiple stages. That means that the tradeoffs among the three do not depend on MS , meaning that $\text{CPI}(\{BB, GB, MA\}, MS)$ holds. Next, the dependency among the triplet $\{BB, GB, MA\}$ is also a result of the option to substitute one by another. As a result, each pair is CPI of the third. Finally, we find that the complementarity of ST and IN is marginal and does not affect the tradeoffs with other attributes. We can therefore verify the following conditions: $\text{CUI}(\{BB, GB, MA\}, MS)$, $\text{CUI}(\{BB, GB\}, MA)$, $\text{CUI}(\{ST, EM, CF, DS, GB, BB\}, IN)$, and $\text{CUI}(\{GB, BB, CF, IN\}, ST)$. The resulting maximal CUI sets for each attribute are shown

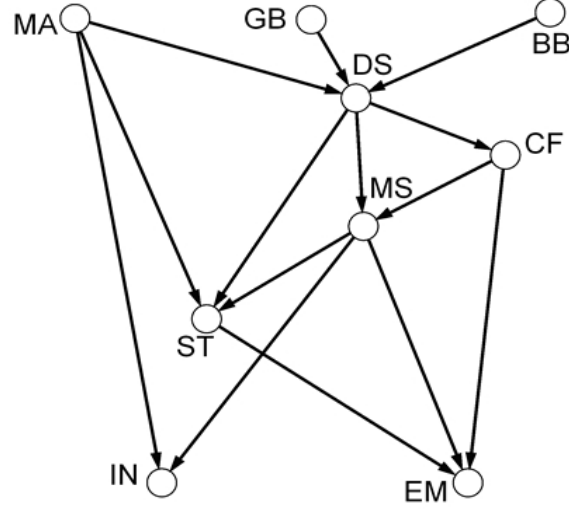


Figure 6.2: An example CUI network. The maximal number of parents is 3, leading to dimension 4.

in Table 6.3.

To construct the network we start with the variable with the largest CUI set, *IN*, which needs only *MS* and *MA* as parents, after which it is *EM* that gets *CF*, *MS*, and *ST* as parents. Next, we consider *ST* which needs four attributes in its conditional set, but *EM* is a descendant, therefore only *DS*, *MS*, and *MA* are needed as parents. The next variable to choose is *MS*, which needs only *CF* and *DS* as parents since the other dependant variables are descendants. Had we chosen *CF* before *MS* it would have needed four parents: *IN*, *MS*, *ST*, and *DS* (note that although *IN* is CUI of *CF* and so is the set $\{BB, GB, MA\}$, this is not the case for the union $\{BB, GB, MA, IN\}$). Now that we choose *CF* after *MS* it has *MS*, *ST*, and *IN* as descendants and therefore only *DS* is a parent. The complete variable ordering is *IN*, *EM*, *ST*, *MS*, *CF*, *DS*, *MA*, *GB*, *BB*, and the resulting CUI network is depicted in Figure 6.2. The maximal dimension is four.

The structure obtained over the utility function in the above example is based largely on objective domain knowledge, and may be common to various sourcing departments. This demonstrates an important aspect of graphical modeling captured by CUI networks: encoding qualitative information about the domain, thus making the process of extracting the numeric information easier. This structure in some cases differs among decision makers, but in other cases (as above) it makes sense to extract such data from domain experts and reuse this structure across decision makers.

6.4 Representation and Elicitation

In this section, I discuss how to elicit utility information from judgments about relative preference differences.

6.4.1 Node Data Representation

Representing $u(\cdot)$ by a CUI network requires that we determine the $f(\cdot)$ and $g(\cdot)$ functions for each CUI condition. At any node y the functions f, g represent the affine transformation of the conditional utility function $u(x^0, Y, Z)$ (here $Z = Pa(x)$) to strategically equivalent utility functions for other values of x . It is shown in Chapter 3 that these transformation functions can be represented in terms of the conditional utility functions $u(x, Y^1, Z)$ and $u(x, Y^2, Z)$ for suitable values Y^1 and Y^2 —as described by Equations (3.1) and (3.2), repeated here for convenience:

$$g(X, Z) = \frac{u(X, Y^2, Z) - u(X, Y^1, Z)}{u(X^0, Y^2, Z) - u(X^0, Y^1, Z)}, \quad (6.14)$$

$$f(X, Z) = u(X, Y^1, Z) - g(X, Z)u(X^0, Y^1, Z). \quad (6.15)$$

6.4.2 Elicitation of Measurable Value Functions

A vNM utility function should be obtained through elicitation of preferences over lotteries, for example using even-chance gambles and their *certainty equivalents* (Keeney and Raiffa, 1976). Based on the preceding discussion, to fully specify $u(\cdot)$ via a CUI network, we need to obtain the numeric values for the conditional utility functions $u(x, Y^1, Pa(x))$ and $u(x, Y^2, Pa(x))$ for each node x . This is significantly easier than obtaining the full n -dimensional function, and can be done using methods described in the preference elicitation literature (Keeney and Raiffa, 1976). In this section I show how elicitation can be conducted in cases when the choice problem is among certain outcomes, but a cardinal representation is nevertheless useful in order to represent willingness-to-pay (Section 3.3).

As already argued in Section 3.3, a simple way to elicit the MVF is by asking the decision maker to provide her WTP to improve from one outcome to another, particularly when these outcomes differ over a single attribute. With this motivation in mind, we first observe from (6.14) that $g(x, Z)$ can be elicited in terms of preference differences, between outcomes that possibly differ over a single attribute.

Furthermore, Theorem 3.1 provides valuable intuition that can help extracting $g(\cdot)$. It

is plausible that we can detect whether x and Y are CAI, complements, or substitutes first, and thus provide constraints on the value of $g(\cdot)$. Furthermore, though both Y and x may depend on Z , in practice I do not expect the *level of dependency* (in terms of complements and substitutes) between Y and x to depend on the particular value of Z . In that case g becomes a single-dimensional function, independent of Z .

$f(x, Z)$, intuitively, is a measurement of WTP to improve from x^0 to x . The value $u(x^0, Y^1, Z)$ is multiplied by $g(x, Z)$ to compensate for the interaction between Y and x , allowing $f(\cdot)$ to be independent of Y . If we perform the elicitation obeying the topological order of the graph, the function $u(x^0, Y^1, Z)$ can be readily calculated for each new node from data stored at its predecessors. First, note that the representation of $f(\cdot)$ in (6.14) allows us to arbitrarily choose any $Y^1 \preceq Y^2$. Choose $Y^1 = Y^0$, and let $Z = \{z_1, \dots, z_k\}$, ordered such that children precede parents. Since Y, x are fixed on the reference point, we can decompose $u(x^0, Y^0, Z)$ based on the CUI network on the subgraph induced by Z , obtaining the following:

$$u(x^0, Y^0, Z) = \sum_{i=1}^k (f_{z_i} \prod_{j=1}^{i-1} g_{z_j}) f_{n+1}(\cdot).$$

Now we can obtain $f(x, Z)$ as follows: first we elicit the preference difference function $e(x, Z) = u(x, Y^1, Z) - u(x^0, Y^1, Z)$. Then, assuming $g(x, Z)$ was already obtained, calculate:

$$f(x, Z) = e(x, Z) - (g(x, Z) - 1)u(x^0, Y^1, Z).$$

6.5 Optimization

One of the primary uses of utility functions is to support optimal choices, as in selecting an outcome or action. The complexity of the choice depends on the specific properties of the environment. When the choice is among a limited set of definite outcomes, we can recover the utility of each outcome using the compact representation and choose the one with the highest value. For instance, in the software example of Section 6.3 we would normally choose among an enumerated set of vendors or packages. In this procurement scenario the utility is an MVF, and we usually choose the outcome that yields the highest utility net of price. In case of decision under uncertainty, when the choice is among actions that lead to probability distributions over outcomes, the optimal choice is selected by computing the expected utility of each action. If each action involves a reasonably bounded number of outcomes with non-zero probability, this again can be done by exhaustive computation.

Nevertheless, it is often useful to directly identify the maximal utility outcome given a

quantitative representation of utility. In case of a direct choice over a constrained outcome space, the optimization algorithm serves as a subroutine for systematic optimization procedures, and such can be adapted from the probabilistic reasoning literature (Nilsson, 1998). The algorithm may also be useful as a heuristic aid for optimization of expected utility or net utility mentioned above, when the set of possible outcomes is too large for an explicit, exhaustive choice.

In this section, I develop optimization algorithms for discrete domains, and show how in many cases CUI networks can provide leverage for optimization of CAI maps. As is typical for graphical models, the optimization algorithm is particularly efficient when the graph is restricted to a tree.

6.5.1 Optimization over CUI Trees

Definition 6.3. A CUI tree is a CUI network, in which the CUI DAG has the property that no node has more than one child.

Note that this type of graph corresponds to an upside-down version of a standard directed tree (or a forest).

Let T be a CUI tree. Assume WLOG that T is connected (a forest can be turned into a tree by adding arcs). As an upside-down sort of tree, it has any number of roots, and a single leaf. I denote the root nodes by $\{a_1, \dots, a_k\}$, the child of $a_i \in \{a_1, \dots, a_k\}$ by b_i , and so on. For each root node a_i , define the function

$$h_{a_i}(b_i) = \arg \max_{a'_i \in D(a_i)} u(b_i, a'_i),$$

denoting the selection of an optimal value of a_i corresponding to a given value of its child. From Proposition 6.5, h_{a_i} does not depend on the reference values chosen for $A \setminus \{a_i, b_i\}$. The function $h_{a_i}(\cdot)$, which I call the *optimal value function* (OVF) of a_i , is stored at node a_i since it is used by its descendants as described below.

Next, each b_i has no children or a single child c_i , and any number of parents. For simplicity of exposition I present the case that b_i has two parents, a_i and a_j . The maximization function for b_i is defined as

$$h_{b_i}(c_i) = \arg \max_{b'_i \in D(b_i)} u(c_i, b'_i, h_{a_i}(b'_i), h_{a_j}(b'_i)).$$

In words, we pick the optimal value of b_i for each assignment to its child and its parents. But since we already know the optimum of the parents for each value of b_i , we need only

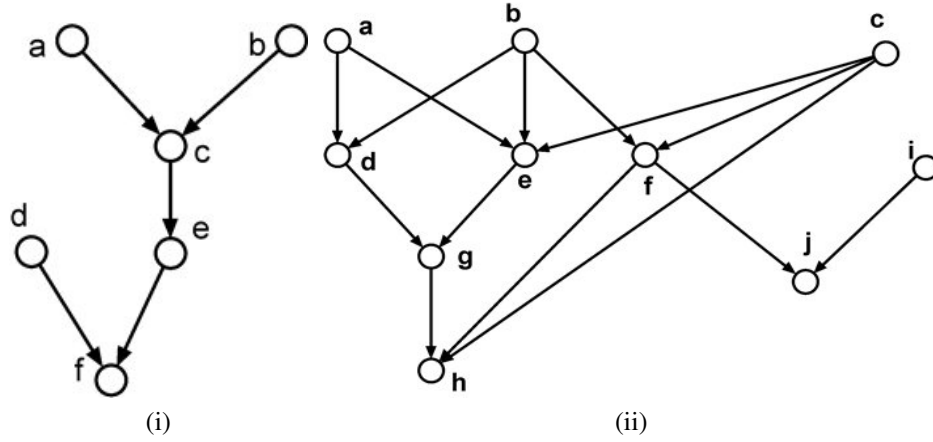


Figure 6.3: CUI networks in optimization examples: (i) Tree (ii) Non-tree

consider this optimum for each evaluation on the domain of b_i .

The only external child of the set $\{a_i, a_j, b_i\}$ is c_i , and it has no external ancestors, hence $\{a_i, a_j, b_i\}$ is CUI of the rest given c_i , therefore the maximization above again does not depend on the reference values of the rest of the attributes. Similarly, when computing $h_{c_i}(d_i)$ for the child c_i of b_i , each value for c_i fixes b_i (and any other parents of c_i), and that fixes a_i and a_j (and the other ancestors of c_i). The last computation, at the leaf x , evaluates each value of x . Each value x' causes this cascade of fixed values to all of the ancestors, meaning we finally get the optimal choice by comparing $|D(x)|$ complete assignments.

I illustrate the execution of the algorithm on the CUI tree of Figure 6.3i. We compute $h_a(c)$ which is the optimal value of a for each value of c , and similarly $h_b(c)$. Next, to compute $h_c(e)$, for each value e' of e we compare all outcomes $(e', c', h_a(c'), h_b(c'))$, $c' \in D(c)$. At node d we compute $h_d(f)$, which is independent of the other nodes. At node e we compute $h_e(f) = \arg \max_{e'} u(f, e', h_c(e'), h_b(h_c(e')), h_a(h_c(e')))$ (node d can be ignored here) and at node f it is

$$h_f() = \arg \max_{f' \in D(f)} u(f', h_e(f'), h_d(f'), h_c(h_e(f')), h_b(h_c(h_e(f'))), h_a(h_c(h_e(f')))).$$

Note how each candidate value of f causes the cascade of optimal values to all of its ancestors. The solution is then $h_f()$ and the resulting values of all the ancestors.

This optimization algorithm iterates over the nodes in topological order, and for each x_i it calculates the OVF $h_{x_i}(x_j)$, where x_j is the child of x_i . This calculation uses the values of the OVF stored for its parents, and therefore involves comparison of $|D(x_i)| |D(x_j)|$ outcomes. In case the numeric data at the nodes is available, factoring in the time it takes to recover the utility value for each outcome (which is $O(n)$), the algorithm runs in time

$$O(n^2 \max_i |D(x_i)|^2).$$

6.5.2 Optimization Over General DAGs

A common way in graphical models to apply tree algorithms to non-trees is by using the *junction graph*. However, the common notion of a junction graph for DAG is a polytree, whereas the algorithm above is specialized to a (unit) tree. Instead, I generalize the tree algorithm in order to optimize the CUI network directly.

In the tree case, fixing the value of the child of a node x is sufficient in order to separate x from the rest of the graph, excluding ancestors. We consider each value of the child at a time, so it also determines the values for all the ancestors. In a general DAG it is no longer sufficient for the OVF to depend on the children, because they do not provide sufficient information to determine the values of $An(x)$. Hence I generalize this notion to be the *scope* of x ($Sc(x)$, defined below), which is a set of nodes on which the OVF of x must depend, in order for an iterative computation of the OVF to be sound.

With this generalization, the DAG algorithm is similar to the tree algorithm. Let G be a CUI DAG, and x_1, \dots, x_n a variable ordering that agrees with the topological order of G (parents precede children). For each x_i (according to the ordering), compute $h_{x_i}(Sc(x_i))$ for any instantiation of $Sc(x_i)$. The optimal instantiation can now be selected backwards from $h_{x_n}()$, since for each node x_i that is reached the values for $Sc(x_i)$ are already selected.

$Sc(x_i)$ is computed as follows: scan variables x_{i+1}, \dots, x_n in this order. When scanning x_j , add x_j to $Sc(x_i)$ if the following conditions hold:

1. There is an undirected path between x_j and x_i .
2. The path is not blocked by a node already in $Sc(x_i)$.

By these conditions, $Sc(x_i)$ includes all the children of x_i , but none of x_i 's ancestor since they precede x_i in the ordering. In addition, $Sc(x_i)$ includes all nodes that are needed to block the paths that reach x_i through its ancestors. For example, if x_k, x_j are children of an ancestor x_a of x_i , and $k < i < j$, then x_j must be in $Sc(x_i)$, because of the path through x_a . The children of x_j are blocked by x_j , so unless they have another path to x_i they will not be in $Sc(x_i)$. The children of x_k , if ordered later than x_i , will be in $Sc(x_i)$ (but their children will not), and so on.

Figure 6.3ii is an example of a CUI network that is not a CUI tree. I consider the scopes under the variable ordering a, b, \dots, j . The scope of roots always equals their set of children (because there is no other path reaching them), meaning $Sc(a) = \{d, e\}$, $Sc(b) = \{d, e, f\}$, $Sc(c) = \{e, f, h\}$, $Sc(i) = \{j\}$. The scope of d must include its child g and its siblings e and f . All paths of h, j , and i to d are blocked by g, e, f therefore $Sc(d) = \{g, e, f\}$. For e , we must include its child g , and its “younger” sibling f . Node h has a blocked path to e through

$f \in Sc(e)$, but also a non-blocked one through $c \notin Sc(e)$, therefore $Sc(e) = \{g, f, h\}$. Similarly, g and h are in the scope of f due to paths through b and c respectively, hence $Sc(f) = \{g, h, j\}$. For g , in addition to its child h we add j whose path to g through f, b, e is not blocked ($Sc(g) = \{h, j\}$) and finally $Sc(h) = Sc(i) = \{j\}$ and $Sc(j) = \{\}$.

The next step, computing the OVF, requires that we compare a set of outcomes that differ on $x_i \cup Co(x_i)$, where $Co(x_i)$ is a set of nodes whose OVF can be determined by $x_i \cup Sc(x_i)$ (hence they are *covered* by x_i). For this maximization to be valid, the condition $CUI(x_i \cup Co(x_i), A \setminus (x_i \cup Co(x_i) \cup Sc(x_i)))$ must hold. I formally define $Co(x_i)$, and establish this result which is proved in the appendix.

Definition 6.4. $Co(x_i)$ is the smallest set of nodes that satisfied the following condition

$$\forall j < i, Sc(x_j) \subseteq (\{x_i\} \cup Sc(x_i) \cup Co(x_i)) \Rightarrow x_j \in Co(x_i). \quad (6.16)$$

Intuitively, x_j is covered by x_i if each node $x_k \neq x_i$ in its scope, is either in the scope of x_i or was determined (according to its own scope) to be covered by x_i . In Figure 6.3ii, $f \in Co(g)$ because $Sc(f) = \{g\} \cup Sc(g)$. $e \in Co(g)$ because $Sc(e) \subset \{g\} \cup Sc(g) \cup \{f\}$. Moreover, $Sc(d) = \{g, e, f\}$ hence $d \in Co(g)$ as well, and similarly we find that $a, b \in Co(g)$. In this example all the nodes preceding g in the ordering are covered, but this is not necessarily always the case.

Lemma 6.7. An assignment to x_i and $Sc(x_i)$ is sufficient to determine $h_{x_j}(\cdot)$ for each $x_j \in Co(x_i)$.

Lemma 6.8. For any node x_i , $CUI(\{x_i\} \cup Co(x_i), A \setminus (\{x_i\} \cup Co(x_i) \cup Sc(x_i)))$. Meaning that x_i and the nodes it covers are CUI of the rest given $Sc(x_i)$.

When the algorithm reaches node x_i , every choice of assignment to $Sc(x_i) \cup \{x_i\}$ determines optimal values for $Co(x_i)$ (Lemma 6.7). We compare the $|D(x_i)|$ assignments which differ over the values of x_i and $Co(x_i)$, and select an optimal one as the value of $h_{x_i}(Sc(x_i))$. This optimum does not depend on the nodes in $A \setminus (\{x_i\} \cup Co(x_i) \cup Sc(x_i))$ due to Lemma 6.8.

To illustrate, examine what happens when the algorithm reaches node g in Figure 6.3ii. At this point $h_x(Sc(x))$ is known for any x that precedes g . As showed, all these nodes are in $Co(g)$. Indeed, the assignment to $Sc(g) = \{g, h, j\}$ directly determines the value for $h_f(\cdot)$, and then together with $h_f(\cdot)$ it determines the value for $h_e(\cdot)$, and further it cascades to the rest of the nodes. The CUI network shows that $CUI(\{a, b, c, d, e, f, g\}, \{i\})$ (given $\{h, j\}$) and therefore the maximization operation (over the choice of value for g) is valid regardless of the value of i .

The performance of the optimization algorithm is exponential in the size of the largest scope (plus one). Note that this would be seriously affected by the choice of variable ordering. Also note, that in the case of a tree this algorithm specializes to the tree optimization above, since there is no node that has a path to an ancestor of x_i , except for other ancestors of x_i which must precede x_i in the ordering. Therefore it is always the case that $Sc(x_i) = Ch(x_i)$, meaning that $h_{x_i}(\cdot)$ is a function of its single child. Based on that, I expect the algorithm to perform better the more similar the CUI network is to a CUI tree.

6.5.3 CUI Tree for Optimization of CAI Maps

The optimization procedure for CUI trees is particularly attractive due to the relatively low amount of preference information it requires. In some cases the comparison can be performed directly, without even having the data that comprises the utility function. Aside from the direct benefit to CUI networks, I am interested in applying this structure to the optimization of CAI maps. In some domains a CAI map is a simple and effective way to decompose the utility function. However, the optimization of CAI maps is exponential in the size of its tree width, and it requires the full data in terms of utility functions over its maximal cliques. If a CAI map happens to have a simple structure, such as a tree, or the CP condition, faster optimization algorithms can be used. However, it could be the case that a CAI map is not a tree, but more subtle CUI conditions might exist which cannot be captured by CAI conditions. If enough such conditions could be detected to turn the CAI map into a CUI tree (or close enough to a tree), we could take advantage of its simple optimization procedure.

Definition 6.5. Let $G = (V, E)$ be a CAI map. An enhanced CAI map is a directed graph $G' = (V, A)$, in which a pair of arcs $(u, v), (v, u) \in A$ implies the same dependency as an edge $(u, v) \in E$, and in addition for any node x , $CUI(A \setminus (\{x\} \cup In(x)), x)$ ($In(x)$ denoting the set of nodes y for which $(y, x) \in A$). I call the pair of arcs $(u, v), (v, u) \in A$ a hard link and an arc $(u, v) \in A$ s.t. $(v, u) \notin A$ a weak link.

For any CAI map, an enhanced CAI map can be generated by replacing each edge (u, v) with the arcs (u, v) and (v, u) . This does not require any additional CUI conditions because these are entailed by the CAI map. However, if additional CUI conditions as above can be detected, we might be able to remove one (or both) of the directions. Figure 6.4i shows a CAI map which contains a cycle. If we could detect that $CUI(\{a, d, f\}, b)$, we could remove the direction (a, b) and get the enhanced CAI map in Figure 6.4ii. The set of CUI

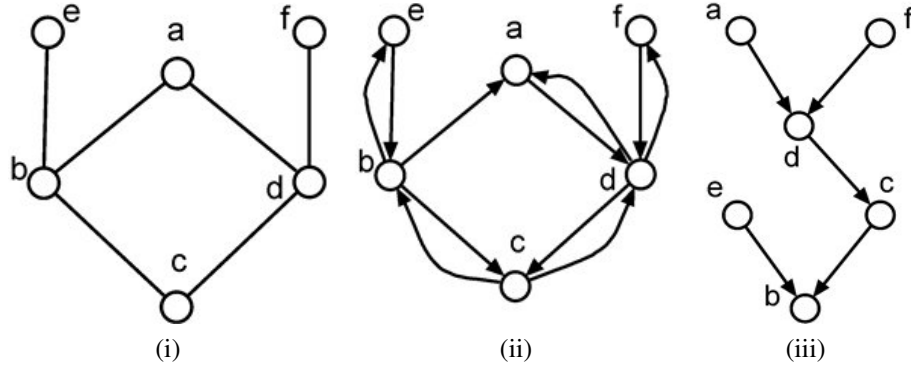


Figure 6.4: (i) A CAI map containing a cycle. (ii) Enhanced CAI map, expressing CUI of $\{a, d, f\}$ in b . (iii) An equivalent CUI tree.

conditions implied by the enhanced CAI map can now be expressed by a CUI tree, as in Figure 6.4iii.

Proposition 6.9. *Consider an enhanced CAI map G . Let ω be an ordering on the nodes of G , and G' a DAG which is the result of removing all arcs (u, v) whose direction does not agree with ω . If for any such removed arc, v is an ancestor of u in G' , then G' is a CUI network.*

For hard links, the removal of (u, v) leaves v as a parent of u , so the condition trivially holds. To obtain a CUI tree, the key is therefore to find a variable ordering under which enough weak links can be removed to turn the graph into a tree, maintaining the condition of Proposition 6.9. For a large number of variables, an exhaustive search over variable orderings may not be feasible. However in many cases it can be effectively constrained, restricting the number of orderings that we need to consider. For example, in order to break the cycle in Figure 6.4ii it is clear that the weak link (b, a) must be implied by the ordering, so that a could be an ancestor of b . The only way for this to happen (given the existing hard links), is that c is a parent of b , d is a parent of c , and a a parent of d .

Proposition 6.10. *Let $c = (y_1, \dots, y_k)$ be a cycle in an enhanced CAI map G . Assume that c contains exactly one weak link: (y_i, y_{i+1}) for some $i < k$, or (y_k, y_1) . Let ω be a variable ordering that does not agree with the order of the path $p = (y_{i+1}, y_{i+2}, \dots, y_k, y_1, \dots, y_i)$. Then any CUI network constructed from G and ω (by Proposition 6.9) is not a tree.*

Therefore any cycle that contains one weak link leads to a constraint on the variable ordering. Cycles with more than one weak link also lead to constraints. If c above has another weak link (y_j, y_{j+1}) , one of the two links must be removed, and the ordering must agree with either the path p above or the path $p' = (y_{j+1}, y_{j+2}, \dots, y_k, y_1, \dots, y_j)$. Assuming

WLOG that $j > i$, the paths (y_{i+1}, \dots, y_j) and (y_{j+1}, \dots, y_i) are required for both p and p' , and therefore can be used as constraints. Similarly we can find the intersection of the paths implied by any number of weak links in a cycle.

Sometimes the constraint set can lead to an immediate contradiction, and in such case search is redundant. If it does not, it can significantly reduce the search space. However, the major bottleneck in preference handling is usually elicitation, rather than computation. Therefore, given that a good variable ordering may lead to the reduction of the optimization problem to a simpler, qualitative task, eliminating the need for a full utility elicitation, it would be worthwhile to invest the required computation time.

6.6 Nested Representation

From Section 6.4.1 I conclude that node data can be represented by conditional utility functions depending on the node and its parents. But this may not be the best dimensionality that can be achieved by a network. Perhaps the set $Z = Pa(x)$ has some internal structure, in the sense that the subgraph induced by Z has maximal dimension lower than $|Z|$. In such a case we could recursively apply CUI decomposition to the conditional utility functions for this subgraph. This approach somewhat resembles the hierarchical decomposition represented by the utility trees (Keeney and Raiffa, 1976; Von Stengel, 1988). For example, to represent f_1 for the network of Figure 6.1, we require the conditional utility function $u(x_1, x_4^1, x_5^1, x_6^1, x_2, x_3)$. However from the network we can see that $CUI(x_3, x_2 \mid x_1, x_4, x_5, x_6)$. Hence we can decompose this conditional utility:

$$u(x_1, x_4^1, x_5^1, x_6^1, x_2, x_3) = f'(x_1, x_2) + g'(x_1, x_2)u(x_1, x_3, x_2^0, x_4^1, x_5^1, x_6^1).$$

I use the notation f' and g' since these are not the same as the f and g functions of the top-level decomposition.

A nested representation can be generated systematically (Algorithm CUI-Nested), by decomposing each local function at node x (x 's *utility factors*) whose argument set $Z \subseteq Pa(x)$ does not form a clique. We do that by performing a complete CUI decomposition over the subgraph induced by Z (keeping in mind that all the resulting factors depend also on x). The algorithm works as follows: Process node in reverse topological order (outermost loop). Decompose each factor stored at current node, whose parents do not form a clique. At each such parent x_r (innermost loop) store resulting new factors. These are defined over x_r , those of x_r 's parents that are also in $Pa(x_i)$ (this is Q_r^d), and a clique A_r^d on which the original factor depends. Each time a factor is decomposed its set

Q shrinks. When it is empty, K is a clique. u_j^i is shown in the proof to be the conditional utility function $u([K_j^i])$.

```

Algorithm:CUI-Nested
Data: CUI Utility factors  $u(x, Pa(x), \hat{Y}), u(x, Pa(x), \tilde{Y})$  at each node  $x$ 
/* note:  $\hat{Y}, \tilde{Y} \in D(Y)$  */
Determine order  $x_1, \dots, x_n$ ;
for  $j = 1, \dots, n$  do /* initialization */
     $K_j^1 = \{x_j\} \cup Pa(x_j)$ ; /* scope of utility factors */
     $Y_j^1 = A \setminus K_j^1$ ; /* rest of variables */
     $Q_j^1 = Pa(x_j)$ ;
     $A_j^1 = \emptyset, d_j = 1$ ;
end
for  $j = 1, \dots, n$  do
    for  $i = 1, \dots, d_j$  do /* loop on factors in node  $j$  */
        if  $Q_j^i \neq \emptyset$  and  $K_j^i$  is not a clique then
            Let  $G_j^i$  be the subgraph induced by  $Q_j^i$ ;
            Decompose  $u_j^i(K_j^i)$  according the CUI network on  $G_j^i$ ;
            foreach  $x_r \in Q_j^i$  do
                Let  $d_r = d_r + 1$  (current num. of factors at  $x_r$ ) and denote  $d = d_r$ ;
                 $A_r^d = A_j^i \cup \{x_j\}, Q_r^d = Pa(x_r) \cap Q_j^i$ ;
                 $K_r^d = A_r^d \cup \{x_r\} \cup Q_r^d, Y_r^d = A \setminus K_r^d$ ;
                Store new CUI factors of  $x_r$ :  $u(K_r^d, \hat{Y}_r^d), u(K_r^d, \tilde{Y}_r^d)$ ;
                /*  $\hat{Y}_r^d, \tilde{Y}_r^d$  are fixed assignments to  $Y_r^d$  */
            end
            Remove factors  $u(K_j^i, \hat{Y}_j^i), (K_j^i, \tilde{Y}_j^i)$ ;
        end
    end
end

```

Proposition 6.11. *Let G be a CUI network for utility function $u(A)$. Then $u(A)$ can be represented by a set of conditional utility functions, each depending on a set of attributes corresponding to (undirected) cliques in G .*

6.6.1 Discussion

This result reduces the maximal dimensionality of the representation to the size of the largest maximal clique of the CUI network. For instance, applying it to the example in Section 6.3 reduces dimensionality from four to three. An important implication is that we can somewhat relax the requirement to find very large CUI sets. If some variables end up

with many parents, we can reduce dimensionality using this technique. As the example below illustrates, this technique aggregates lower-order CUI conditions to a more effective decomposition.

The procedure may generate a complex functional form, decomposing a function multiple times before the factors become restricted to a clique. The ultimate number of factors required to represent $u(A)$ is exponential in the number of such nesting levels. However, each decomposition is based on a CUI network on a subgraph, and therefore typically reduces the number of entries that are maintained.

I expect the typical application of this technique to be for *composition* rather than *decomposition*. We execute Algorithm CUI-Nested without the actual data, resulting in a list of factors per node (that are conditional utility functions over cliques of the graph). That means that for elicitation purposes we can restrict attention to conditional utility functions over maximal cliques. Once these are obtained, we have sufficient data for all the factors. We can then recover the original, more convenient CUI-network representation of the function and store it as such (more on that in the example below). Therefore, the effective dimensionality for elicitation is that of the maximal cliques. The storage for efficient usage requires the potentially higher dimension of the original CUI network, but typically this is less of a concern.

With this result and Proposition 6.6, CUI networks are shown to always achieve weakly better dimensionality than CAI maps, since both representations reduce the dimensionality to the size of the maximal clique.

6.6.2 Example

I illustrate this result using a simple example. Consider a domain with four attributes (a, b, c, d) , and the following CUI conditions:

$$\text{CUI}(b, c), \text{CUI}(c, b), \text{CUI}(d, a)$$

The CUI network corresponding to the variable ordering a, b, c, d is depicted in Figure 6.5. Since the CUI sets are small (a single variable each), for any variable ordering there must be a node with two parents, meaning dimensionality of three. The nesting operation below combines these lower-order conditions to reduce the dimensionality to two.

Initially, the utility function is represented using the conditional utility functions listed according to their corresponding nodes in the column “Level 0” in Table 6.4. To remove the three-dimensional factors, we need to decompose the functions of node a according to

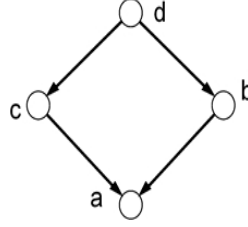


Figure 6.5: Nesting example

the CUI network on $\{b, c\}$, which contains no arcs. This proceeds as follows:

$$u(a, b, c, d^1) = f_b^1(a, c) + g_b^1(a, c)u(a, b, c^0, d^1) = f_b^1(a, c) + g_b^1(a, c)(f_c^1(a, b) + g_c^1(a, b)u(a, b^0, c^1, d^1))$$

$$u(a, b, c, d^2) = f_b^2(a, c) + g_b^2(a, c)u(abc^0, d^2) = f_b^2(a, c) + g_b^2(a, c)(f_c^2(a, b) + g_c^2(a, b)u(a, b^0, c^0, d^2))$$

The resulting functions are $f_b^i(a, c)$, $g_b^i(a, c)$, $f_c^i(a, b)$, $g_c^i(a, b)$, $i = 1, 2$. The functions $f_b^i(a, c)$ and $g_b^i(a, c)$ can be represented using the conditional utility functions $u(a, b^1, c, d^i)$ and $u(a, b^2, c, d^i)$, and similarly the other two functions. We can delete the factors of a , $u(a, b, c, d^i)$, and add the new lower dimensional factors as a second column to a 's parents b and c . Though we had to multiply the number of factors we store by four, all the new factors are conditional utility functions over subdomains of the (deleted) higher dimensional factors. The algorithm continues to node b , and loops over its six factors. If there are factors that are defined over a set of parents of b that are not a clique it decomposes them and store the new factors in the next table column. In case a factor from column “level 1” could be decomposed, we would add a “level 2” column to store the result. In my simple example no further decomposition is possible.

Attr	Level 0 (CUI net)	Level 1
a	$u(a, b, c, d^1), u(a, b, c, d^1)$	
b	$u(a^0, b, c^1, d),$ $u(a^0, b, c^2, d)$	$u(a, b, c^1, d^1), u(a, b, c^2, d^1)$ $u(a, b, c^1, d^2), u(a, b, c^2, d^2)$
c	$u(a^0, b^1, c, d),$ $u(a^0, b^2, c, d)$	$u(a, b^1, c, d^1), u(a, b^2, c, d^1)$ $u(a, b^1, c, d^2), u(a, b^2, c, d^2)$
d	$u(a^0, b^0, c^0, d)$	

Table 6.4: Nested CUI decomposition.

The reverse direction mentioned above is done as follows. We run Algorithm CUI-Nested without any data, resulting in a table such as Table 6.4 (without the actual utility values). We then elicit the data for the non-deleted factors (all are limited to maximal cliques). Next, we recover the more convenient “level 0” CUI representation using the table, by computing each deleted factor (going from rightmost columns to the left) as a function of the factors stored at its parents.

6.7 Summary

I present a graphical representation for multiattribute utility functions, based on conditional utility independence. CUI networks provide a potentially compact representation of the multiattribute utility function, via functional decomposition to lower-dimensional functions that depend on a node and its parents. CUI is a weaker independence condition than those previously employed as a basis for graphical utility representations, allowing common patterns of complementarity and substitutivity relations disallowed by additive models.

I propose techniques to obtain and verify structural information, and use it to construct the network and elicit the numeric data. In addition, I develop an optimization algorithm that performs particularly well for the special case of CUI trees. In some cases it can also be leveraged for efficient optimization of CAI maps. Finally, I show how functions can be further decomposed over the set of maximal cliques of the CUI network. With this technique, CUI networks can achieve the same dimensionality of graphical models based on CAI and GAI decompositions, yet with more broadly applicable independence conditions.

Chapter 7

Modeling Preferences for Sponsored Search

This chapter presents another application of preference structures in electronic trading. *Sponsored search auctions* are conducted by search engines to select the advertisements that appear beside the results shown to a user in response to a search query. Sponsored search is an emerging market yielding billions of dollars of search-engine revenues per year.

7.1 Introduction

The most popular search engines, Google, Yahoo, and MSN Live, currently rank and price ads using an auction mechanism, which works as follows. Advertisers place per-click bids on search keywords that are relevant to their ads. Each time a search user types in a query, all the advertisers that have bids that match the query are then sorted using a ranking function, and the top advertisers have their ads shown, according to the order of the sort, along with the results returned by the search engine. An advertiser is charged only if the search user clicks on his ad, in which case the user is directed to the advertiser's web site, called *the landing page*. The common implementation of this auction, called *generalized second price* (GSP), uses a generalization of the pricing rule of the well-known second-price auctions: the per-click charge to each winning advertiser is the minimum bid needed to retain his position in the ad ranking.

In most analyses of the GSP auction, the ranking function is assumed to be either the per-click bids, or the “per impression” bids obtained by multiplying each per-click bid by the *click-through rate* (CTR) of the corresponding ad. The latter method is sometimes referred to as ranking “by revenue”. I refer to GSP auctions using these ranking functions as GSP_B and GSP_R , respectively.

GSP_B and GSP_R may be appropriate for extracting short-term revenue from the advertisers, but they do not represent the value of the ads to the search user. This could result in displaying of low-quality ads, that can hurt the search experience, and that in turn can lead users to switch to another search engine. Moreover, having negative experiences may cause users to click less on ads in the future. Both these outcomes may reduce the long-term revenues of the search engine.

In GSP_R , the problem of low-quality ads is mitigated to some degree by having the click-through rate in the ranking function: the lower the ad CTR, the higher the per-click cost to the advertiser. Despite this, the quality problem remains for GSP_R for at least two reasons. First, a low CTR advertisement may still be displayed if there is no sufficient competition for the particular search query. Second, advertisers can maintain high CTRs by submitting ads that *appear* relevant to the query, regardless of whether or not a typical user clicking on the ad will find the resulting web site relevant to his search. Such ads, whose cost is reduced due to their higher CTR, can be profitable for advertisers who for various reasons gain from increasing traffic volume to their web site.

Search engines have existing methods for reducing the number of low-quality ads. Several use a ranking function that includes a quality measure, but the specifics regarding how this measure is obtained, or how it is used are not disclosed.

In this chapter, I study principled methods for incorporating ad quality into a ranking function. The two main contributions in this topic are as follows. On the theoretical side, I design a family of utility functions to represent user preferences over ads. I use constructions from multiattribute utility theory that allow these functions to be represented using a reasonably small number of parameters, while still allowing for rich preferences to be expressed. Next, I show how the search-engine publisher can make explicit trade-offs between user utility and short-term revenue, leading to a mechanism that is similar to GSP, but its ranking function includes the utility of the user. The other contribution is in two types of simulations I perform. Short-term simulations show the loss of short-term revenue required to improve user satisfaction under various parameterizations of the system. Then, I simulate the effect of the ranking system on long-term revenues through its impact on user satisfaction. Both types of simulations allow to evaluate the various utility models I suggest.

The chapter is structured as follows. First, I present the utility-theoretic framework (Section 7.3) and a ranking system that deals with the conversion between user's utility and revenue (Section 7.5). The second part describes the simulation analysis. In Section 7.7, I present short-term, static simulations of the new ranking system. In Section 7.8, I build a model of long-term user interaction with the system and provide simulation results.

7.2 Related work

There is a growing body of literature that discusses sponsored search mechanisms, most of which considers various variations of the GSP mechanism. Varian (2007) defines a Nash Equilibrium concept for GSP auctions, based on the equilibrium of the corresponding one-shot static game of complete information. In equilibrium bidders cannot improve their value by acquiring a different slot. That means that their payoff from increasing their bid to improve their position, or their payoff from reducing their bid to achieve a lower position, cannot exceed their current payoff.

Varian further introduces a more specific notion of equilibrium, called *symmetric Nash Equilibrium* (SNE). SNE holds if bidders cannot gain from bidding the current price of positions either below or above their current position. In SNE, the GSP outcome is efficient, in the sense that bidders are ranked according to their valuations per per-click (in GSP_B) or impression (in GSP_R).

A variation of symmetric equilibrium was also introduced by Edelman et al. (2007), who use the term *locally envy-free* equilibrium. They describe a generalized English auction that converges to an outcome of a locally envy-free equilibrium (which is also shown to be bidder-optimal), and might indicate that bidding converges to this outcome in practice.

Both these authors generalize their results to the case that slots are not allocated directly according to bid amount, but rather using GSP_R (or, in general, a product of the bid and a quality measure of the ad). Lahaie (2006) analyzes the properties of GSP with ranking in either methods. Börgers et al. (2006) generalize the efficiency implication of SNE to a more general model, in which bidders' click valuation varies according to the position of their ad. In this work I stay within Varian's framework in assuming that click value is position independent.

Particularly relevant to this work are several papers that explicitly consider the user as a player in the mechanism, or consider the user's experience as an objective. Lahaie and Pennock (2007) consider the click-through rate of an ad as an indicator of its relevance to the user's search. In their model, ads are ranked according to a product of the bid by click-through rate, but the CTR is raised to the power of some $q \in [0, 1]$, so it captures both rank-by-bid and rank-by-revenue as special cases. They suggest specifying a minimum acceptable level of user's experience, indicated by the extent to which CTR is considered in the ranking (i.e., the level of q), and optimize q within the acceptable range using a simulation technique. In contrast, my starting point is that the CTR may not be an accurate indication to the relevance of the ads, because advertisers may have incentives to misrepresent their ad as more relevant to the query than the resulting landing page actually

is.

Abrams and Schwarz (2007) take a more specific approach to user’s experience, and model the user’s utility from a click as a *hidden cost*. Athey and Ellison (2007) suggest the following model for their equilibrium analysis: an advertiser’s value from a click corresponds to the probability that the ad meets the consumer’s need. Consumers scan ads from top to bottom, and click until an ad meets their need or until the cost of clicking exceeds the expected value of clicking. In the simulations, I adopt Athey and Ellison’s modeling of the user’s click decision according to computation of expected utility. In contrast to both Athey and Ellison and Abrams and Schwarz, I focus on a principled method to model the user’s utility, based on available or computable information such as the relevance of the landing page to the query, and use it as a practical ranking method. Further, I perform simulations that incorporate the user’s trust in the system as an adaptive parameter that affects the expected utility of a click.

7.3 Modeling User Experience

In this section, I describe models for search-user utility as a function of the interactions that the user has with the ads that are shown. Initially, I limit the discussion to models that are *additive* over these interactions; that is, the utility for the interactions on a set of ads Λ is simply the sum of the utility for the interaction on each individual ad $\lambda \in \Lambda$.

Let $\lambda \in \Lambda$ be a particular ad shown to the user. I define the following four mutually exclusive and collectively exhaustive interactions events $I(\lambda)$ that the user can have with this ad:

$I(\lambda) = i_s$ The user looks at λ but does not click (“scan”).

$I(\lambda) = i_{cg}$ The user looks at λ , clicks, and finds the resulting page relevant (“good”).

$I(\lambda) = i_{cb}$ The user looks at λ , clicks, and does not find the resulting page relevant (“bad”).

$I(\lambda) = i_{ig}$ The user does not look at λ (“ignored”).

In this model, a user must look at an ad before clicking it; hence whenever a user clicks on an ad, we know the interaction is either i_{cg} or i_{cb} .

I postulate the existence of a vNM utility function over interaction events, $U : I(\lambda) \rightarrow \Re$. Note that the utility $U(i)$ is defined over the four possible events, independently of a particular ad λ . A natural assumption is that the maximum value \bar{u} for $U(i)$ occurs when $i = i_{cg}$

and that the minimum value \underline{u} occurs when $i = i_{cb}$. The function $U(\cdot)$ is scaled and normalized, without loss of generality, such that $\bar{u} - \underline{u} = 1$ and $U(i_{ig}) = 0$. As a result the function has two free parameters: $U(i_s)$, and either $U(i_{cb})$ or $U(i_{cg})$.

7.4 Publisher Utility Model

Search users currently do not participate in any monetary transfer with the search engine, and therefore it is up to the publisher to incorporate user utility into the search mechanism through the choice of which ads are shown. In this section, I show how to construct a publisher ranking function that makes an appropriate tradeoff between user utility and revenue.

7.4.1 Publisher Value of User Utility

Let $v(u, r)$ denote the publisher's value, in dollars, of providing utility u to the user and receiving short-term revenue r . There are two simple ways that might be reasonable to consider the tradeoff between u and r . In a *multiplicative* function, the publisher is willing to lose a fixed fraction t_m of the revenue in order to improve the user utility from the minimum value to the maximum value; that is, $v(\underline{u}, r) = v(\bar{u}, (1 - t_m)r)$. In an *additive* publisher utility function, a publisher is willing to lose a fixed absolute amount t_a in order to improve the user utility from the minimum value to the maximum value; that is, $v(\underline{u}, r) = v(\bar{u}, r - t_a)$. Note that each indifference constraint is satisfied by a family of functions.

There are advantages to each version of the publisher utility function, and we can get the advantages of both by assuming a multi-linear form for $v(u, r)$ that accommodates both a multiplicative and an additive indifference. The combined indifference constraint is as follows:

$$v(\underline{u}, r) = v(\bar{u}, (1 - t_m)r - t_a). \quad (7.1)$$

That is, the publisher is willing to spare a fraction t_m of the revenue and an additional t_a dollars from the remainder in order to increase user utility from the minimum \underline{u} to the maximum \bar{u} . By specifying a non-zero t_m , the publisher indicates that he values the user experience in high-stake searches more than the experience in searching for a cheap item. On the other hand, if the publisher is always willing to spare a constant small amount t_a in order to improve a user's utility, then for low-revenue auctions he can improve user utility dramatically with low cost.

A simple multi-linear form that satisfies (7.1), given that $\bar{u} - \underline{u} = 1$, is

$$v(u, r) = t_a u + t_m (u - \underline{u})r + (1 - t_m)r. \quad (7.2)$$

This form is consistent with assuming that the publisher’s preferences exhibit mutual utility independence of u and r (Definition 2.11). In the experiments, I consider publisher utility functions of the form of (7.2).

7.4.2 Probabilistic Model

A utility maximizing publisher will rank sets of ads by expected (publisher) utility given a particular query. For the purpose of computing this expectation, I model the probability distribution over the interaction events in this section.

I assume that the interaction events for a set of ads are mutually independent, and thus the joint probability for all events decomposes into a product of probabilities $P_{\lambda,k}(\cdot)$ that are specific to each ad λ and a position k ¹. The independence assumption is violated if the probability of the user clicking on one ad depends on the quality and/or quantity of other ads clicked. More accurate models can be estimated from data; for the purpose of this work, however, I keep the independence assumption for simplicity.

User-interaction events decompose into up to three sub-events: the user looks at the ad or not, the user clicks on the ad or not, and a clicked ad is relevant or not. As common in the sponsored-search-auction literature (Edelman et al., 2007; Varian, 2007; Börger et al., 2006; Lahaie, 2006), I assume that the click probability α_λ^j of an ad λ in position j is the product of an ad effect π_λ , and a positional effect β^j , so that $\alpha_\lambda^j = \pi_\lambda \beta^j$. Both the positional effect β^j and the ad-specific factor π_λ are computed based on data.² In the model the positional effect is interpreted as the probability that the user looks at the ad (“scan”). To complete the probabilistic model, we need the probability ρ_λ that the landing page of an ad is relevant to the query given that the ad was clicked. In the experiments, I computed ρ_λ as follows: I used a computed similarity score between the text of the ad-destination page and the text of the organic search-result summaries shown in the results page, and converted this score to a probability using a calibration model constructed from human-labeled data.³

Putting these pieces together, now $P_{\lambda,j}(i_s) = \beta^j \cdot (1 - \pi_\lambda)$ (the user looked at the ad but

¹Note that $P_{\lambda,k}(\cdot)$ will certainly depend on the given query; this dependence is left implicit to simplify notation.

²The positional effect can be estimated by observing click rates of the same ad shown in different positions. In the experiments I used a shared positional model for every query.

³The development of the similarity score algorithm, the collection of human-labeled data, and the statistical analysis of the data were all performed by Microsoft AdCenter.

did not click), $P_{\lambda,j}(i_{cg}) = \beta^j \cdot \pi_\lambda \cdot \rho_\lambda$ (the user looked at the ad, clicked, and the landing page was relevant), $P_{\lambda,j}(i_{cb}) = \beta^j \cdot \pi_\lambda \cdot (1 - \rho_\lambda)$ (the user looked, clicked, and the landing page was not relevant), and $P_{\lambda,j}(i_{ig}) = 1 - \beta^j$ (the user did not look at the ad).

7.4.3 Publisher Ranking Function

I now show how to compute the expected publisher utility for a set of ads by combining the models from the previous two subsections.

Given an advertisement λ with bid b_λ in position j , the publisher's expected utility $v^e(\lambda, b_\lambda, j)$ is computed by summing over the interaction events:

$$v^e(\lambda, b_\lambda, j) = \sum_i P_{\lambda,j}(i) v(U(i), r(i, b_\lambda))$$

The short-term revenue r that results from a click is the advertiser's bid b , hence b replaces r when (7.2) is computed per-click. Hence if i is a click event (i_{cg} or i_{cb}) then $r(i, b_\lambda) = b_\lambda$. Because under GSP the price is the lowest bid needed to retain the given position, it follows that using the bid in the publisher utility function will rank ads in the same order as if we had used the true revenue (i.e., the price). If i is not a click event (i_s or i_{ig}), then $r(i, b_\lambda) = 0$.

Now each of the utility terms in the sum above is expanded using the publisher-utility model (7.2). When $i = i_{cg}$, the publisher utility is

$$v_{cg}(b_\lambda) = t_a U(i_{cg}) + [t_m(U(i_{cg}) - \underline{u}) + (1 - t_m)] \cdot b_\lambda,$$

and similarly when $i = i_{cb}$, the publisher utility is

$$v_{cb}(b_\lambda) = t_a U(i_{cb}) + [t_m(U(i_{cb}) - \underline{u}) + (1 - t_m)] \cdot b_\lambda.$$

When $i = i_s$, the publisher utility is $t_a U(i_s)$ (there is no short-term revenue), and when $i = i_{ig}$ the user utility and the revenue are both zero and thus the publisher utility is zero. Combining with the probability model, and using the definitions v_{cg} and v_{cb} above, we have:

$$v^e(\lambda, b_\lambda, j) = \beta^j (1 - \pi_\lambda) t_a U(i_s) + \beta^j \pi_\lambda (\rho_\lambda v_{cg}(b_\lambda) + (1 - \rho_\lambda) v_{cb}(b_\lambda)). \quad (7.3)$$

A utility-maximizing publisher will allocate ads so as to maximize the sum of (7.3) over all positions. Without loss of generality, assume that the positional effects β^j are monotonically decreasing with j (i.e., the click rate of a particular ad will be higher if it is placed higher in the list). We can achieve the utility-maximizing allocation by using a *position*

independent ranking function $\mu(\lambda, b_\lambda)$ to greedily assign ads to positions, starting from $j = 1$ and continuing until all slots are filled or until the best unassigned ad has a negative value.

$$\mu(\lambda, b_\lambda) = \frac{v^e(\lambda, b_\lambda, j)}{\beta^j} = (1 - \pi_\lambda)t_a U(i_s) + \pi_\lambda(\rho_\lambda v_{cg} + (1 - \rho_\lambda)v_{cb}) \quad (7.4)$$

The equivalence of ranking by (7.3) and by (7.4) follows from the fact that the total expected utility $\sum_j v^e(\lambda, b_\lambda, j)$ for a set of ads is the dot product of the vector of $\mu(\cdot)$ values for each position and the corresponding vector of positional effects. In particular, because the positional effects are monotonically decreasing with j , the total expected utility is maximized by placing ads with highest value of $\mu(\cdot)$ in the positions with the highest positional effects.

Furthermore, we can use a reserve price to guarantee that prices are such that we only show ads whose expected utility is positive. The same argument as above, based on the properties of GSP, ensures that using the bid b is equivalent to using the price in terms of deciding what ads to show.

7.5 Sponsored Search Mechanism

In the previous section, I derived a position-independent ranking function that allows publishers to incorporate user utility into the selection of ads. Note that the ranking function from (7.4) can be expressed as follows:

$$\mu(\lambda, b_\lambda) = f_\lambda + g_\lambda b_\lambda. \quad (7.5)$$

That is, by expanding the definitions of $v_{cg}(b_\lambda)$ and $v_{cb}(b_\lambda)$ and grouping terms as appropriate, the ranking function can be expressed as a linear function of the bid value. In this section, I study properties of the GSP mechanism when using any ranking function of the form of (7.5). Note that this class of ranking function includes as special cases the two common ranking functions described in Section 7.1—ranking by bid is achieved with $f_\lambda = 0$ and $g_\lambda = 1$, and ranking by revenue is achieved with $f_\lambda = 0$ and $g_\lambda = \pi_\lambda$.

I now provide a formal definition of GSP. Let Φ denote the candidate ads for a particular query. Let $B_\Phi = \{b_\lambda \mid \lambda \in \Phi\}$ denote the corresponding set of bids. Let $\mu(\lambda, b_\lambda)$ be an arbitrary position-independent publisher ranking function. The sorted series resulting from applying $\mu(\cdot)$ on B_Φ is denoted by Λ_μ , and its j th element by $\Lambda_\mu(j)$. Finally, I use m to denote a reserve price.

Definition 7.1. GSP_μ is an allocation and pricing scheme that allocates slot j to $\lambda = \Lambda_\mu(j)$, for all $j \leq M$ for which $\mu(\lambda, b_\lambda) \geq m$. The price paid by λ is p_λ^j such that

$$\mu(\lambda, p_\lambda^j) = \max(\mu(\phi, b_\phi), m), \quad (7.6)$$

where $\phi = \Lambda_\mu(j+1)$.

That is, GSP_μ ranks by $\mu(\cdot)$ and charges a bidder the minimum price required to remain in his position. Let $\tilde{\Lambda}_\mu$ denote the set of slots allocated by $\mu(\cdot)$; note that $\tilde{\Lambda}_\mu$ is Λ_μ truncated to the number of slots actually allocated. When μ represents expected publisher utility, a natural choice for m here is zero, ensuring that an ad is displayed only if it is rational to do so.

When $\mu(\cdot)$ is of the linear form of (7.5), we use GSP_L to denote the corresponding mechanism. Based on (7.6), the price assigned by GSP_L to the slot j won by λ is

$$p_\lambda = \frac{f_\psi - f_\lambda + g_\psi b_\psi}{g_\lambda}, \quad (7.7)$$

where $\psi = \Lambda_\mu(j+1)$.⁴

7.5.1 Equilibrium Properties

Properties of GSP have been analyzed in previous literature (Lahaie, 2006; Edelman et al., 2007; Varian, 2007; Börgers et al., 2006). I mainly follow the settings and definitions of Varian (2007), which analyzes GSP_B but also considers the Google variation, in which $\mu(\lambda, b_\lambda) = q_\lambda b_\lambda$ for some unknown quality factor q_λ .⁵ GSP_L is different from this variation of GSP due to the additive element that is included in the ranking formula.

I denote the *valuation*—the amount by which a bidder truly values a click—for the advertiser of λ by h_λ . I follow Varian’s complete-information setup: because traders can adjust their bid at any time, a stable set of bids can be seen as an equilibrium of a static game in which traders know each other’s bids. In all of the following discussion, the case in which the price is determined by m is ignored for simplicity. Further, s and t each refer either to a position in $\tilde{\Lambda}_\mu$ or to a losing position (i.e., a position $j > |\tilde{\Lambda}_\mu|$ for which $\beta^j = 0$). I use Λ_B in place of Λ_μ to denote the result of ranking by GSP_B .

⁴More accurately, the price is the maximum over p_λ and m .

⁵Recently, Yahoo! has started using an unknown quality factor as well.

Definition 7.2 (Varian (2007)). Let $\lambda = \Lambda_B(s)$. A set of bids is a Nash Equilibrium (NE) in GSP_B if for all $s \neq t$,

$$\alpha_\lambda^s(h_\lambda - p_\lambda) \geq \alpha_\lambda^t(h_\lambda - p_\psi), \quad (7.8)$$

where for $t < s$, $\psi = \Lambda_B(t)$, and for $t > s$, $\psi = \Lambda_B(t + 1)$.

In words, the advertiser for each ad λ is making more total return in his current position than he would by adjusting his bid to move to another position. The asymmetric conditions under (7.8) reflect the fact that in order for λ to be moved to a *higher* ranked position t , the bid needs to exceed that of the current holder of position t , whereas to move to a *lower* ranked position t , the bid need only beat the bid of the current holder of $t + 1$.

I now adapt Definition 7.2 to the more general linear ranking function of (7.5). To emphasize the ranking function in the notation, I use $L(\lambda, b_\lambda)$ in place of $\mu(\lambda, b_\lambda)$ and I use Λ_L in place of Λ . The main change to the definition results from the fact that the per-click prices depend on the advertiser.

Definition 7.3. Let $\lambda = \Lambda_L(s)$. A set of bids is a Nash Equilibrium (NE) in GSP_L if for all $s \neq t$,

$$\alpha_\lambda^s(h_\lambda - \frac{f_\phi - f_\lambda + g_\phi b_\phi}{g_\lambda}) \geq \alpha_\lambda^t(h_\lambda - \frac{f_\psi - f_\lambda + g_\psi b_\psi}{g_\lambda}) \forall t,$$

where $\phi = \Lambda_L(s + 1)$ and for $t < s$, $\psi = \Lambda_L(t + 1)$, and for $t > s$, $\psi = \Lambda_L(t)$.

Varian (2007) also introduces *symmetric Nash equilibrium* in which ψ in Definition 7.2 refers to $\Lambda_B(t + 1)$ regardless of the relative values of s and t . This more restrictive equilibrium has the advantage that it leads to a tractable computation of equilibrium bids. Generalizing the definition to GSP_L yields:

Definition 7.4. Let $\lambda = \Lambda_L(s)$. A set of bids is a symmetric Nash equilibrium (SNE) in GSP_L if for all $s \neq t$,

$$\alpha_\lambda^s(h_\lambda - \frac{f_\phi - f_\lambda + g_\phi b_\phi}{g_\lambda}) \geq \alpha_\lambda^t(h_\lambda - \frac{f_\psi - f_\lambda + g_\psi b_\psi}{g_\lambda}),$$

where $\phi = \Lambda_L(s + 1)$ and $\psi = \Lambda_L(t + 1)$.

The following observation, which is similar to the one used by Varian in discussing the Google variation of GSP, shows that through the ranking function $L(\cdot)$ there is a simple one-to-one mapping between equilibria in GSP_L and GSP_B .

Theorem 7.1. Let $B_\Phi = \{b_\lambda \mid \lambda \in \Phi\}$ be a bids profile, and let $\hat{B}_\Phi = \{L(\lambda, b_\lambda) \mid \lambda \in \Phi\}$. Further, let $H_\Phi = \{h_\lambda \mid \lambda \in \Phi\}$ be the valuations of the advertisers, and $\hat{H}_\Phi = \{L(\lambda, h_\lambda) \mid \lambda \in \Phi\}$. Then: (1) B_Φ is NE in GSP_L with valuations H_Φ if and only if \hat{B}_Φ is NE in GSP_B with valuations \hat{H}_Φ . (2) B_Φ is SNE in GSP_L with valuations H_Φ if and only if \hat{B}_Φ is SNE in GSP_B with valuations \hat{H}_Φ .

This mapping allows us to easily adapt results about equilibrium points in GSP_B to results about equilibrium points in GSP_L . Using this technique, and given known results on GSP_B in SNE (Varian, 2007), it can be established that GSP_L not only optimizes publishers utility with respect to bids, but also with respect to true valuations:

Theorem 7.2. Let B_Φ be SNE bids in GSP_L with valuations H_Φ . Then, out of all rankings over Φ , Λ_L maximizes

$$\sum_{j=1}^{|\Lambda_L|} (f_{\Lambda_L(j)} + g_{\Lambda_L(j)} h_{\Lambda_L(j)}) \beta^j. \quad (7.9)$$

When $L(\cdot)$ is defined using publisher utility as in (7.4), we see that the optimized term is precisely the total expected publisher's utility, with the true value h_λ in place of the bid b_λ . Intuitively, Proposition 7.2 shows that in equilibrium GSP_L “does the right thing” with respect to publisher utility. More precisely, following Lahaie (2006), we call a ranking that would have been selected if we knew the true bidders' valuations a *standard allocation*, and this result shows that in SNE, GSP_L selects a standard allocation.

7.5.2 Welfare Maximization

Proposition 7.2 can be seen as a generalization of efficiency results of GSP_B and GSP_R (Varian, 2007). When we restrict GSP_L to be GSP_R , that is $f_\lambda = 0$ and $g_\lambda = \pi_\lambda$, the maximized term can be seen as the total welfare of the advertisers. If the publisher is assumed to have no value for the ads, then SNE in GSP_R is efficient, meaning it maximizes total welfare. A natural question here is whether we can say something similar about GSP_L . Is there an interpretation of $v^e(\cdot)$ under which Proposition 7.2 suggests some kind of efficiency?

To this end, we should first reformulate the problem as a welfare maximization problem, rather than publisher's utility maximization. That means we should reinterpret $v^e(\cdot)$ as an expected total welfare. We stay faithful to our interpretation of user utility: it is incorporated into the mechanism, represented as the publisher's welfare, and its monetary equivalence is expressed through $v(u, r)$. We cannot, however, interpret directly $v^e(\cdot)$ as the sum of the publisher's and the advertisers' welfare because $v(u, r)$ exhibits a multiplicative dependency between u (user utility/publisher's welfare) and r (advertisers' reported

welfare). Fortunately, we observe that the multiplicative term only captures the fact that the publisher attributes higher value to higher-stake searches, and does not express real dependency in any specific bid. Rather, this term captures a general revenue potential of the query. We can modify slightly the publisher's utility such that $v(u, r)$ is a sum over the revenue r and the monetary equivalence of utility, where the latter does not depend on r but rather on some other value \hat{r} that indicates the revenue potential of the query:

$$\hat{v}(u, r) = t_a u + (t_m(u - \underline{u}) - t_m)\hat{r} + r \quad (7.10)$$

in which $t_a u + (t_m(u - \underline{u}) - t_m)\hat{r}$ represents the monetary value of user utility. We can derive sensible values for \hat{r} by using the bids of similar queries or by using the bids of the same query from previous instances. We can now replace all instances of $v(u, r)$ in (7.3) by $\hat{v}(u, r)$. Thus (7.9), which sums over $v^e(\cdot)$, represents true total welfare because the bids are replaced with true valuations. Proposition 7.2 can be proved for this modified mechanism as well (omitted), which shows efficiency of GSP_L in SNE.

This is of course, efficiency only with respect to the equilibrium behavior. Moreover, it is only the case in SNE, which is a restricted subset of NE. We also note that the study by Börger et al. (2006) found no evidence that SNE is likely to occur in practice.

7.6 Model with Nonadditive Utility

Until this point I ignored potential dependencies among the user utilities from different ads in the same impression. This assumption may be too restrictive, if we expect the utility from an ad to change depending on what kind of experience the user had with other ads in the page. For example, the negative impact of a second irrelevant ad may be viewed as either stronger or weaker than the impact of the first irrelevant ad; furthermore, this impact may be different depending on whether the user also encounters a relevant ad. I discuss these potential implications in detail in Section 7.6.2.

To relax this assumption, we should consider the expected utility of a *set of ads*. For a given set of ads Λ , I assume that the user has exactly one interaction with each ad. Let $n_s(\Lambda)$ denote the *number* of ads in Λ for which the user has interaction $I = i_s$, and similarly $n_{cg}(\Lambda)$, $n_{cb}(\Lambda)$ and $n_{ig}(\Lambda)$ denoted the number of ads for which the other interaction events occurred. When Λ is clear from context, I use n_s, n_{cg} , n_{cb} and n_{ig} without arguments to simplify notation.

Consider a user who is shown five ads $\Lambda = \{\lambda_1, \dots, \lambda_5\}$ with his search results. Suppose the user ignores the first ad, looks at the next two ads (without clicking), clicks on

the fourth ad and finds it irrelevant, and then clicks on the fifth ad and finds useful information. Then the five interaction events are $I(\lambda_1) = i_{ig}$, $I(\lambda_2) = i_s$, $I(\lambda_3) = i_s$, $I(\lambda_4) = i_{cb}$, and $I(\lambda_5) = i_{cg}$, and the corresponding counts are $n_s(\Lambda) = 2$, $n_{cg}(\Lambda) = 1$, $n_{cb}(\Lambda) = 1$, and $n_{ig}(\Lambda) = 1$.

Another assumption is that the user's utility for a set of ads Λ depends only on the space Θ of triplets $\langle n_s, n_{cg}, n_{cb} \rangle$, and the utility function $u : \Theta \rightarrow \Re$, is a vNM utility function. Each possible series of ads Λ leads to a known probability distribution (lottery) over the outcomes (the space Θ), and the utility is taken with expectation over this space. More specifically, for a given query we obtain (Section 7.6.1) a probability distribution $P_\Lambda : \Theta \rightarrow [0, 1]$ —I leave the dependence on the query implicit to simplify notation. We compute the expected utility $u^e(\Lambda)$ for Λ by summing over all interaction-event counts that sum to $|\Lambda|$:

$$u^e(\Lambda) = \sum_{\substack{n_{cg}+n_{cb}+ \\ n_s+n_{ig}=|\Lambda|}} u(n_{cg}, n_{cb}, n_s) P_\Lambda(n_{cg}, n_{cb}, n_s, n_{ig}). \quad (7.11)$$

Note that the utility function does not depend on the number of ads for which $I = i_{ig}$; if a user does not look at an ad, then that ad plays no role in the utility of the set shown.

7.6.1 Extending the Probabilistic Model

To simplify notation in this section, I use $P(\cdot)$ instead of $P_\Lambda(\cdot)$ throughout. To derive the model $P(n_{cg}, n_{cb}, n_s, n_{ig})$ over the set of event *counts*, I first consider the probability of any sequence of interaction events that a user can have with an ordered sequence of ads. More specifically, given an ordered sequence of ads $\Lambda = (\lambda_1, \dots, \lambda_K)$, we construct an *event* model $P^e(\{I(\lambda_1), \dots, I(\lambda_K)\})$ that represents the probability of each of the 4^K interaction sequences. Note that the dependence of this model on the query and actual sequence of ads is again left implicit. As detailed below, we will obtain $P(\cdot)$ for a given set of counts by summing $P^e(\cdot)$ over all event sequences consistent with those counts.

I retain the probabilistic independence assumption over the ads; each interaction event in the sequence depends only on the position and identity of the corresponding ad:

$$P^e(\{I(\lambda_1), \dots, I(\lambda_K)\}) = \prod_{k=1}^K P_{\lambda_k, k}^e(I(\lambda_k)). \quad (7.12)$$

Given the assumption of Eq. (7.12), we can construct the count model $P(n_{cg}, n_{cb}, n_s, n_{ig})$ incrementally as follows. Let $P_k(n_{cg}, n_{cb}, n_s, n_{ig})$ denote the count model for the first k elements of Λ , and thus we have that $P(\cdot) = P_{|\Lambda|}(\cdot)$. We can derive P_k from P_{k-1} by summing

over the possible interaction events for the k th element of Λ . Letting λ denote this k th element we have:

$$\begin{aligned}
P_k(n_{cg}, n_{cb}, n_s, n_{ig}) = & \\
& P_{k-1}(n_{cg} - 1, n_{cb}, n_s, n_{ig}) \cdot P_{\lambda,k}^e(i_{cg}) \\
& + P_{k-1}(n_{cg}, n_{cb} - 1, n_s, n_{ig}) \cdot P_{\lambda,k}^e(i_{cb}) \\
& + P_{k-1}(n_{cg}, n_{cb}, n_s - 1, n_{ig}) \cdot P_{\lambda,k}^e(i_s) \\
& + P_{k-1}(n_{cg}, n_{cb}, n_s, n_{ig} - 1) \cdot P_{\lambda,k}^e(i_{ig})
\end{aligned} \tag{7.13}$$

Note that in Equation (7.13), the sum of the event counts $n_{cg} + n_{cb} + n_s + n_{ig}$ is necessarily equal to k . Also, if any of the counts are negative, then the corresponding probabilities in the recursion are defined to be zero. The base case for the recursion is for $k = 0$, corresponding to an empty sequence of ads, for which $P_0(0, 0, 0, 0)$ is one. The term (7.13) can be computed efficiently using dynamic programming, that is starting from $P_0(0, 0, 0, 0)$ and increasing k at each iteration.

7.6.2 Utility Models

Throughout this section, I use superscript integers to denote the assignment of a specific numeric value to an argument to $u(\cdot)$ (e.g., I use n_{cg}^1 to denote $n_{cg} = 1$). This is also used in notation such as $u([n_{cg}^1])$, in which $n_{cg}^0, n_{cb}^0, n_s^0$ serves as the reference point. $u([n_{cg}^1])$ is therefore a shorthand for $u(n_{cb} = 0, n_{cg} = 1, n_s = 0)$. With some abuse of notation, I omit the brackets because the full argument list is always the same, that is $u(n_{cg}^1)$ is used instead of $u([n_{cg}^1])$. $u(\cdot)$ is normalized such that $u(n_{cg}^0, n_{cb}^0, n_s^0) = 0$, which means that seeing no ads results in zero utility. Normally, the marginal utility of a relevant clicked ad will be positive, and the marginal utility of an irrelevant clicked ad or scanned ad will be negative. An irrelevant clicked ad, called here a “bad” ad, is usually more costly to the user’s experience than a merely scanned ad.

The additive model analyzed above takes the assumption that $u(n_{cg}, n_{cb}, n_s)$ is constrained to the following linear form:

$$u(n_{cg}, n_{cb}, n_s) = n_{cg}u(n_{cg}^1) + n_{cb}u(n_{cb}^1) + n_s u(n_s^1). \tag{7.14}$$

In words, each ad contributes a constant additive utility depending on its interaction type.

I first obtain Equation (7.14) and other forms of $u(\cdot)$ to follow more formally from specific MAUT assumptions. In the discussion that follows, I use N to denote the set of count attributes $\{n_{cg}, n_{cb}, n_s\}$ that define the outcome space Θ . First, consider the assump-

tion that the set of attributes $N = \{n_{cg}, n_{cb}, n_s\}$ is additive independent (Definition 2.13). If we further restrict each of the (one-dimensional) functions in the additive representation (Equation 2.4) to be linear, we get Equation (7.14). I denote by *LIN* the family of utility function that are restricted to this form.

In the simulations, I use as a benchmark a particular member of the *LIN* family in which the utility is the accumulated relevance probability of the clicked ads. I name this specific utility function *UER* for “utility equals relevance”. If we use the *UER* model, the contribution of each ad λ to the user’s expected utility simplifies to the product of the (landing page) relevance and CTR for λ . In addition, for the simulation purposes I define two additional linear utility models. *LIN-LC* (“low cost”), in which the value of a clicked relevant (“good”) ad ($u(n_{cg}^1)$) exceeds the cost of a clicked irrelevant (“bad”) ad ($u(n_{cb}^1)$) by about 50%, and *LIN-HC* (“high cost”), where the cost of a bad ad is 25% higher than the value of a good ad.

The *LIN* form requires the assessment of only two parameters. As mentioned above, it may be too restrictive if we expect that the utility of an ad depends on its context. For example, the negative impact of a second irrelevant ad may be viewed as either stronger or weaker than the impact of the first irrelevant ad; in other words, we might have decreasing or increasing margins over n_{cb} . Similarly, the utility we get from clicking on the first relevant ad may be higher than the utility we get from the second relevant ad, and so on.

The margins of an attribute may also be affected by the value of a different attribute. For example, we may be less annoyed from irrelevant clicked ads when we also experience relevant clicked ads. Or, in contrast, the user’s experience over a set of ads may resemble a bag of apples, in which one rotten instance ruins all, meaning it is enough that one irrelevant ad is clicked to eliminate the positive experience from relevant ads. I refer to utility models that accommodate the first dependency (and are linear otherwise) as *LCBG* (less cost of bad ads given good ads), and to the one rotten apple model by *ORA*.

With this kind of dependency we can no longer assume additive independence as in Definition 2.13. We can, however, still take advantage of preference structure. For example, the cost of scanning and the cost of clicking irrelevant ads may still be additive, for any fixed value of n_{cg} . We can therefore apply conditional additive independence (Definition 2.14). The assumption $\text{CAI}(n_{cb}, n_s \mid n_{cg})$ means that preferences over lotteries depend only on their margins over $\{n_{cb}, n_{cg}\}$ and $\{n_s, n_{cg}\}$. This condition results in the following functional decomposition (Eq. 2.5):

$$u(n_{cg}, n_{cb}, n_s) = u(n_{cg}, n_s) + u(n_{cg}, n_{cb}) - u(n_{cg}). \quad (7.15)$$

This allows to capture the dependency between n_{cg} and n_{cb} . The two-dimensional functions $u(n_{cg}, n_{cb})$ and $u(n_{cg}, n_s)$ may still be decomposed by using the weaker UI condition (Definition 2.9). I postulate that $UI(n_{cg}, n_{cb})$.⁶ As mentioned before, UI is not necessarily a symmetric condition. Under the ORA model, clicking on an irrelevant ad is much more costly when the user also clicked on a relevant ad, and this can be shown to violate $UI(n_{cb}, n_{cg})$.

The assumption $UI(n_{cg}, n_{cb})$ leads to the following decomposition, based on Equation 2.1:⁷

$$u(n_{cg}, n_{cb}) = u(n_{cb}) + \frac{u(n_{cg}^1, n_{cb}) - u(n_{cb})}{u(n_{cg}^1)} u(n_{cg}) \quad (7.16)$$

This means that instead of the two-dimensional function $u(n_{cg}, n_{cb})$, we need to assess three one-dimensional functions, $u(n_{cg})$, $u(n_{cb})$, and $u(n_{cg}^1, n_{cb})$. To express an ORA model we can set $u(n_{cb}^1) = u(n_{cb}^i)$ to a low (high negative) value, and set $u(n_{cg}^1, n_{cb}^1) = u(n_{cg}^1, n_{cb}^i)$ to be only slightly better. To express LCBG, we set $u(n_{cg}^1, n_{cb}) - u(n_{cb}) > u(n_{cg}^1)$ for $n_{cb} > 0$ (and the difference gets bigger the larger n_{cb} is). This way the marginal contribution of n_{cg} gets larger with more bad ads, so it offsets some part of their cost.

The cost of the scanned ads n_s reflects, in addition to the loss of screen space, potential annoyance of the user from having paid ads in the search results page. If the user clicks some ads and gets useful information, this part of the cost of the scanned ads may be mitigated. That means that the marginal utility of n_s may be higher (less negative) when $n_{cg} > 0$. For this reason, I do not assume that CAI holds between n_{cg} and n_s . Nevertheless, the condition $UI(n_s, n_{cg})$ may hold. We do value the scanned ads at a higher cost when $n_{cg} = 0$, but this may not affect the *preference order over lotteries* over n_s .

In addition we may assume $UI(n_{cg}, n_s)$ (whether ads were scanned or not does not affect the preference over lotteries on n_{cg}). The two-directional independence is in fact mutual utility independence (Definition 2.11) and it therefore leads to the following form, which is a special case of Equation 2.3.

$$u(n_{cg}, n_s) = u(n_{cg}) + u(n_s) + Ku(n_{cg})u(n_s) \quad (7.17)$$

I define a utility model that accommodates lower cost for scanned ads given a good ad (LCSG). Such a model can be achieved by a negative value for K : when $u(n_{cg}) > 0$, some of the negative utility of scanned ads cancels out.

The result of the above assumptions is that we can accommodate models such as ORA, LCBG, and LCSG and still represent $u(\cdot)$ in terms of single dimensional functions: $u(n_{cb})$,

⁶Technically, we say that $CUI(n_{cg}, n_{cb} \mid n_s)$.

⁷For an exact derivation of this form see proof of Theorem 3.2 in appendix.

$$u(n_{cg}, n_{cb}, n_s) \stackrel{(7.15)}{=} \left\{ \begin{array}{l} u(n_{cg}, n_{cb}) \stackrel{(7.16)}{=} u(n_{cb}) + \frac{u(n_{cg}^1, n_{cb}) - u(n_{cb})}{u(n_{cg}^1)} u(n_{cg}) \\ \oplus u(n_{cg}, n_s) \stackrel{(7.17)}{=} \oplus \left\{ \begin{array}{l} u(n_{cg}) \\ u(n_s) \\ Ku(n_{cg})u(n_s) \end{array} \right. \\ -u(n_{cg}) \end{array} \right.$$

Figure 7.1: Functional decomposition of $u(n_{cg}, n_{cb}, n_s)$.

$u(n_{cg}^1, n_{cb})$, $u(n_{cg})$, $u(n_s)$ (and the constant K). More generally, I believe that these assumptions are reasonable and the resulting models can accommodate the most significant dependencies that we would expect to find in a user's utility function. Figure 7.1 shows the series of decompositions described above.

To further reduce the assessment burden, I assume that $u(n_s)$ and $u(n_{cg})$ are linear, and that $u(n_{cb})$ and $u(n_{cg}^1, n_{cb})$ are quadratic, which results in at most six free parameters to any one of the utility models. To complete the specification of explicit versions of these models, for simulation purposes, I fixed the free parameters using subjective assessment.

7.6.3 Mechanism for Nonadditive Utility

A utility maximizing publisher would like to select a set of ads Λ that maximizes the expected value of $v(u, r)$. When the utility is not additive across the ads, we must consider all possible combinations of ads in order to achieve the optimal outcome. To simplify the computation, and to retain the GSP pricing methodology, I define the mechanism *GSP-G*, which is a greedy, heuristic simplification: the ad placements are computed incrementally, such that for each position j we pick the optimal ad given that the set of ads Λ for positions $1, \dots, j-1$ are determined. For this purpose, the ranking function computes the marginal publisher's utility $v^e(\Lambda_{j-1}, \lambda, j)$ of each candidate ad λ for the next position j based on its expected revenues and its effect on the overall user utility. This effect is the *marginal expected user utility* of each candidate, given the ads in previous spots, and it is computed as the difference in the value of (7.11) with and without the candidate. The price can then be defined according to the GSP principle: advertiser of λ that is selected for position j pays the minimum bid he had to submit to win this position, given the ads already selected for positions $1, \dots, j-1$.

Note that because under GSP the price is the lowest bid needed to retain the given position, it follows that this ranking function will rank ads in the same order as if we had used the true price instead of b_λ in $v^e(\cdot)$. Furthermore, setting $v^e(\Lambda_{j-1}, \lambda, j) = 0$ as the reserve price ensures that the choice of how many ads to show is also done based on true revenue.

The greedy approach biases in favor of the higher positions, which is a desired effect because the ads in higher position might have more impact on the user's experience, merely because they are positioned higher. GSP-G is used in the simulations.

7.7 Short-Term Simulations

In the first type of simulations I performed, I took a set of bids for a given query and re-ranked them using GSP-G. This allowed me to test the effect on expected revenues, the relevance of displayed ads, and how it handles “bad” ads, that is ads that have low relevance and high CTR. In the next section I simulate the effect of ranking with user's utility on the long-term revenues.

I used real advertisers' bids submitted to Microsoft Live Search, for a set of about 1000 queries, and a total of 7500 bids. I retrieved up to 8 bids per query, and limited the number of ads shown to 6. I measured two types of ratio: the improvement in relevance to loss of revenue, and the appearance of bad ads to loss of revenue. These ratios cannot capture the subtleties of the user experience as I model it, but nonetheless they can help the publisher find the right tradeoff in terms of the values of t_a and t_m . In addition, I compared the performance of different utility functions. In Figures 7.2i and 7.2ii I measure the ratio of total expected revenue of the ads (as sum of prices multiplied by CTR) to relevance. The publisher makes an explicit choice of the tradeoff between user utility and bids; the results, however, are measured directly in terms of relevance and actual prices.

I calculated the relevance score by shifting the relevance probability to be between -0.5 to 0.5 , so that ads more likely to be irrelevant had a negative effect on the score. I aggregated ad relevance across positions using the information retrieval notion of discounted cumulative gain (DCG) (Järvelin and Kekäläinen, 2000). In all of the simulations I tested the ratio obtained by varying t_a from 0 to 99 cents.

Figure 7.2i compares the results of using LIN-LC and UER, using several choices of t_m (0, 0.2, 0.4, 0.8). First note the deep slope on the right end of each line, which corresponds to t_a varying from 0 to 30 – 40 cents. This shows how using the additive factor allows significant gains in relevance for small amounts of revenue loss, verifying my expectations based on the discussion in Section 7.5. The choice of t_m shifts the whole series up and to

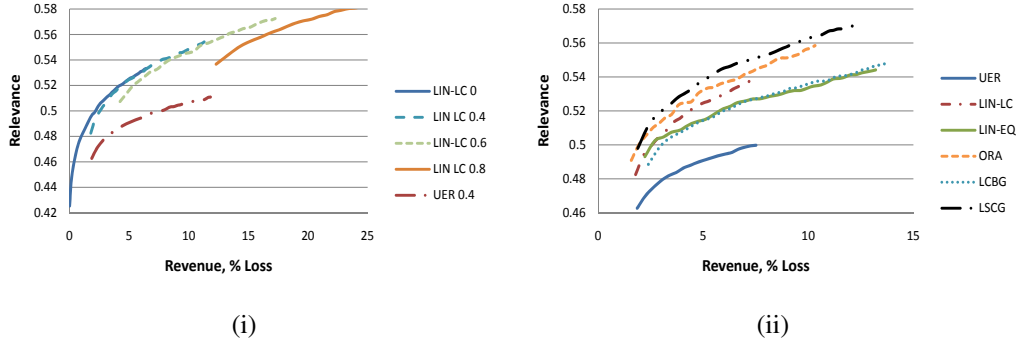


Figure 7.2: Revenue loss for improvement of relevance: (i) using LIN (ii) other utility functions, with $t_m = 0.4$

the left, that is it leads to a more drastic loss of revenue for higher overall relevance. It does not significantly affect the ratio of relevance to revenue except that for high t_m we must use higher values for t_a as well or else the ratio would be lower.

Figure 7.2ii compares relevance vs. revenue across the example utility models, with t_m fixed to 0.4, and in addition shows ORA and LIN-LC with $t_m = 0$ for comparison. As evident from both figures, the simple model UER is clearly dominated by the others. This is due to the fact that it never assigns negative utility, and therefore always shows M ads or the maximum available. The other example utilities do not show some of the less relevant ads, resulting in a better score, and better ratio despite the revenue loss from the dropped ads. When LIN was hard-coded to show all the ads, even when they yield negative publisher utility, it achieved a slightly lower ratio than UER. Also note that LSCG exhibits a better ratio than LIN because it has lower scanning costs in some cases. The scanning cost improves utility in a way that is not measured by Figure 7.2ii, because it is not affected by relevance. If we believe that this cost is indeed less significant when the user's search need is met by the ad, this function allows to achieve the improved revenue to relevance ratio without hurting overall utility.

I consider the experience of clicking on an irrelevant ad particularly costly, therefore, I am interested specifically in those ads which combine high CTR and low relevance. These ads promise a higher revenue potential, but with a greater cost to user experience. Because they are likely to be “bad” and counted in n_{cb} , I refer to them as “bad ads”. When using appropriate utility functions, bad ads may be removed, demoted to a lower position, or remain in their position but pay a higher price. Each of these has different implications on revenue, and I examined these implications as affected by different utility functions.

I defined a bad ad as an ad λ with relevance $\rho_\lambda < 0.45$ (1197 bids out of a total of 7504), and CTR $\pi_\lambda > 0.1$. That selected close to a half of the bids (3451). The intersec-

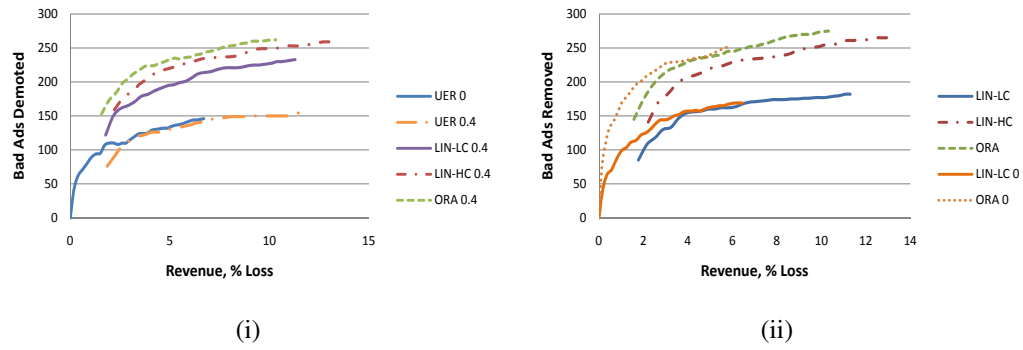


Figure 7.3: Revenue concession for: (i) demotion of bad ads, assuming all ads are shown, (ii) removal of bad ads.

tion, containing 350 bids, defined the set of bad ads. With these definitions and with about six ads per impression, a user views an attractive but irrelevant ad roughly once in three search queries. Figure 7.3i shows how many bad ads were demoted when I used each of the utility functions with $t_m = 0.4$ (t_a varies as above). UER is clearly dominated, because it does not target bad ads specifically, while ORA achieves a somewhat better ratio than the others. Here too, I point out that the additive tradeoff term improves performance. For example, using ORA and $t_m = 0$, between 80 and 100 bad ads are demoted with hardly any effect on revenue. As evident from the charts of ORA and LIN-LC for $t_m = 0$ and $t_m = 0.4$, this effect cannot be achieved by t_m on its own.

Bad ads were removed, by being demoted below the top six positions, or sometimes by showing less ads. The ratio of bad ad removal to revenue was similar for all utility functions, except for UER (Figure 7.3ii). For example, 218 bad ads were removed by ORA with $t_a = 30$ and $t_m = 0.4$, with revenue loss of 3.2%. Additionally, all utility functions increase the average price paid by a bad ad that has not been demoted.

I conclude this section by describing a sample test point. LSCG with $t_m = 0.4$ and $t_a = 39$, achieved relevance score (DCG) of 0.175 (compared to 0.083 of GSP_R), shows 3.6 ads on average (instead of 5.11), removes 61 bad ads, and demotes 48. The other 58 bad ads pay on average 2 cents more. This was achieved for a sacrifice of 6% of the short-term revenues.

7.8 Long-Term Simulations

My basic postulation is that by incorporating user's utility into the ranking, the publisher may be sacrificing a portion of the short-term revenues, but increasing the potential for

long-term revenues due to the greater trust that the users will have in the system. In this section, I describe a dynamic simulation method in which I build a simple model of how users react to clicking on relevant and irrelevant ads, and how their future behavior changes accordingly. The dynamic simulations serve two purposes: first, to show that based on several reasonable assumptions, using a ranking system that incorporates user utility can lead to higher long-term revenues. Second, it can be used to help the publisher choose the system parameters t_a and t_m .

The basic framework is as follows: a large number of search agents are querying the search system. For each query, the system returns a set of ads using a ranking system currently tested. An agent gets a set of ads, and decides if and which ads to click on. Those ads could be either relevant or not. Irrelevant ads decrease the trust of the agent in the system, and with it the probability that he will click on ads in the future.

The experimental setup is as follows. I obtained bid data for a set of about 4000 arbitrarily chosen queries (not the same set used for the static simulations). In each simulation, I chose a ranking system (with values for t_m, t_a and a utility type) and launched a large number of artificial agents (100,000). In each iteration, each agent selects a query at random and obtains a set of ranked ads from the system. The agent scans and clicks on some of the ads according to the model detailed below. The number of iterations that I use varies. In particular, I do short-term simulations (500 iterations) and long-term simulations (2500 iterations). I compare the results of various simulations with respect to the revenue per iteration and the total revenue.

7.8.1 Click Model

Agent i gets a set of results for a given query. It loops over the ads, and clicks on an ad in current position j if the following two conditions hold:

1. The agent scanned the ad.
2. The expected utility of clicking the ad, according to the user's utility function, is positive.

I consider the act of scanning as “looking at the ad with intention to click on ads that seem relevant”. To check the first condition, we need the *probability of scanning* an ad in position j . This measure depends on the query, because the more commercially oriented the query is, the more likely a user is to consider clicking on ads. The CTR of a pair (query, ad) is a product of the probability of scanning and the probability that the ad is clicked given that it was scanned. In the model the second term indicates the probability that a user considers the snippet of the ad relevant. As before, I model the position dependent component of the CTR, that is β^j in $\alpha_\lambda^j = \beta^j \pi_\lambda$, as a query specific *probability of scanning*, and now π_λ is

the probability that the snippet is relevant.

Clearly, the probability of scanning (e.g. for position 1) for a query q cannot be lower than the highest CTR of the set of ads for q , because in each click the ad must have been scanned. Moreover, I make the assumption that there is at least one ad with a relevant snippet in each set of results. That means that for the ad λ^* with the highest CTR in the set, $\pi_{\lambda^*} = 1$. As a result β^j cannot be larger than $\alpha_{\lambda^*}^j$. From these two properties we get that $\beta^j = \alpha_{\lambda^*}^j$ and $\pi_{\lambda} = \frac{\alpha_{\lambda}^j}{\alpha_{\lambda^*}^j}$. Therefore, to decide whether agent i scans ad j the system flips a coin with probability $\alpha_{\lambda^*}^j$. Note that if an agent scans ad $j - 1$ and does not click, the probability of scanning ad j is only the positional difference between $j - 1$ and j , meaning $\frac{\beta^j}{\beta^{j-1}}$.

To compute the expected utility of a click, given that the ad is scanned, we need to define the user's belief over the ad's relevance. Two signals affect this belief: the relevance of the snippet π_{λ} , and the history of ads that user encountered. The effect of the history can be regarded as a measurement of the trust of the user in the system, or the probability, in the user's eyes, that if the snippet is relevant, the landing page would be relevant too. Let ω_i denote this belief. User i 's belief that the landing page of ad λ is relevant (ρ_{λ}^i) is a product of the probability that the snippet is relevant, and the probability that the landing page is relevant given that the snippet is relevant:

$$\rho_{\lambda}^i = \pi_{\lambda} \omega_i.$$

Given a user's utility model, we can now compute the expected utility $\mathcal{E}_u^i(\lambda)$ for user i from a click on λ . If the utility model is additive over the ads, the expected utility computation need only consider the current ad. The expectation is taken over the two possible outcomes of the click, that is that the landing page is relevant (i_{cg}) or not (i_{cb}):

$$\mathcal{E}_u^i(\lambda) = \rho_{\lambda}^i u(n_{cg}^1) - (1 - \rho_{\lambda}^i) u(n_{cb}^1).$$

For nonadditive utility functions, the search agent remembers the number of good, bad and scanned ads resulted this far from the current query (n_{cg}, n_{cb}, n_s respectively), and determines the expected utility of the next ad as follows:

$$\mathcal{E}_u^i(\Lambda_{j-1}, \lambda) = u(n_{cg} + 1, n_{cb}, n_s) \rho_{\lambda}^i + u(n_{cg}, n_{cb} + 1, n_s) (1 - \rho_{\lambda}^i) - u(n_{cg}, n_{cb}, n_s). \quad (7.18)$$

Note that we deduct the utility obtained so far to get the marginal effect of the next ad.

It is left to discuss how the user computes and maintains ω_i . ω_i represents the likelihood that the system will return a relevant page, given a relevant snippet. In other words, it is the belief about the correlation between perceived relevance and real relevance. The user clicks on an ad when he sufficiently believes that the ad is relevant, therefore we can determine ω_i by comparing the number of ads that the user clicked and were relevant to those that the user clicked and were not relevant. I model this belief as a beta distribution over these parameters. Let α_i denote the number of relevant ads that user i encountered, and β_i denotes the number of irrelevant ads. I define $\omega_i = \frac{\alpha_i}{\alpha_i + \beta_i}$, the mean of the beta distribution. α and β are initialized in the beginning of the simulation, as described next. After each click, the system flips a coin according to the computed probability of relevance of the landing page, to determine whether the ad was relevant or not, and increments α or β accordingly.

An important simulation parameter is the initial pair α, β , that is the initial belief of the user, and his confidence in that belief. When the user has an initial confident belief that the system is trustworthy, the default ranking system will do well for a longer period of time. Using the click model above, I found that the default ranking system returns a relevant ad with probability 0.6 (six out of ten ads have $p_\lambda > 0.5$). In order to model transition from the default system to a new one, each agent's α parameter is drawn uniformly in the range $[0, 120]$, and the β parameter is drawn from $[0, 80]$. Using other values lead to different results over a small number of iterations, but the results are not qualitatively different over a large number of iterations.

The simulations do not model the possibility that advertisers respond to the new ranking system and change their bid. I cannot rule out the possibility that the advertisers will respond in a way that changes the results. However, it is reasonable to assume that advertisers would be lead to improve the relevance of their landing page, and perhaps mitigate the discrepancy between the snippet relevance and the real relevance, both of which are desired effects.

7.8.2 Varying Tradeoff Knobs

In the first experiment, I used various combinations of t_a and t_m , and compared how the amount of revenue achieved by each combination changed over time. In all the experiments of this section I used LIN-LC as the utility function for the ranking system, and assumed that agents' utility function is LIN-LC as well. The combination of $t_a = 0$ and $t_m = 0$ is equivalent to the default ranking system GSP_R . Figure 7.4i shows the results during 400 iterations. Initially, the default ranking system provides higher revenue per iteration, but that quickly changes in favor of the new ranking system. Interestingly, the combination

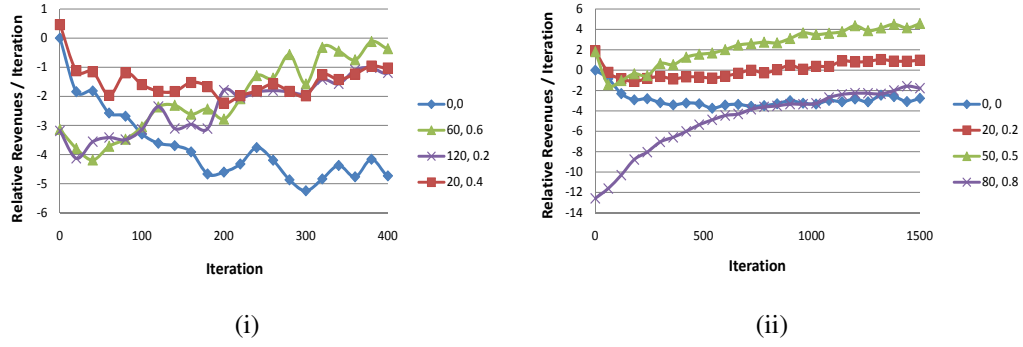


Figure 7.4: Revenue each iteration, using various t_a, t_m , with different initial beliefs. (i) $\alpha = 600, \beta = 400$ (ii) $\alpha = 60, \beta = 40$. Series names are t_a, t_m . “0,0” refers to GSP_R .

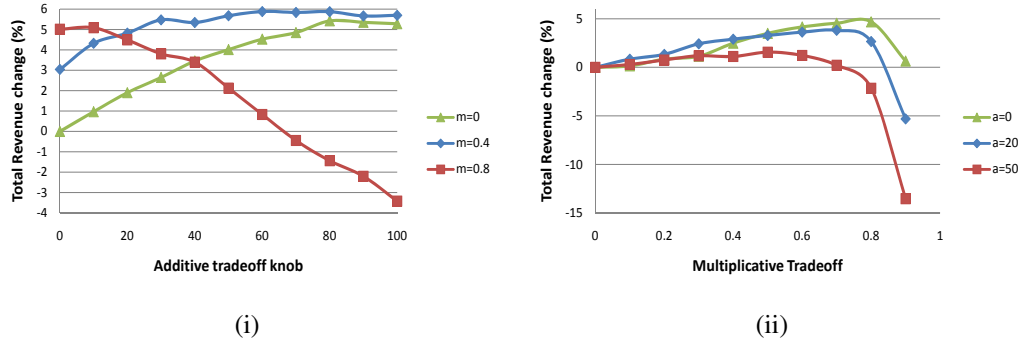


Figure 7.5: Total revenue for 2500 iterations, varying t_a and t_m . (i) varying t_m , with t_a between 0 and 50. m stands for t_m . (ii) varying t_a , with t_m between 0 and 0.99. a stands for t_a .

with $t_a = 20, t_m = 0.4$ yields higher revenue than the default already in the first iteration, meaning that the effect of better ranking leads to more clicks within the first impression.

Figure 7.4ii shows similar results, with the same mean, but a higher initial count: α is drawn (uniformly) from $[0, 1200]$ and β from $[0, 800]$.

In Figures 7.4i and 7.4ii I used a few sample combinations of tradeoff knobs. Next, I performed a more systematic study to optimize this choice. I compared the total revenue achieved after 1500 iterations, when fixing the value of t_m and varying t_a (Figure 7.5i), and the other way round (Figure 7.5ii). Evidently, it is beneficial to increase t_a up to some point, after which the long-term revenues start to decline slightly. The exact value of this point depends on t_m : for example, with $t_m = 0.4$, a value of 30 for t_a is sufficient. Note that the starting point of the series $t_m = 0$ represents the performance of the default system.

7.8.3 Varying Utility Functions

The key question I examine in this section is whether having a system that models the utility function used by the users results in higher long-term revenue than a system that does not model it correctly. The relevance model is not sufficient to capture the effect of the utility model associated with a user. For example, if the user considers the cost of clicking a bad ad higher than the value of seeing a good ad, it results in a more cautious click model, but the result of the click in terms of user experience are not fed back into the model. As describe next, I introduce this effect through the relevance count. This means that the relevance count is now interpreted more broadly, as a general user satisfaction measure.

I first apply this idea to LIN-LC and LIN-HC. To model the effect of the utility on the satisfaction, each good ad encountered is added to the count of relevant ads, weighed by the utility of a single good ad. Each bad ad encountered is added to the count of irrelevant ads, weighed by the utility of a single bad ad. I performed two experiments. First, I compared a system that uses LIN-LC with a system that uses LIN-HC given that the agents are modeled with LIN-HC. Next, I compared the the two systems given that the agents are modeled by LIN-LC. The results are after 2500 iterations (given $t_a = 20, t_m = 0.5$) indicate that the system performs better when the utility is modeled correctly. Given LIN-HC users, the revenue achieved by a LIN-HC system was 2.5% higher than the revenue achieved by a LIN-LC system. Given LIN-LC users, the revenue achieved by a LIN-LC system was slightly higher than the one achieved by LIN-HC system. When the tradeoff knobs are lower, the impact of using the correct utility function is of course weaker, but the qualitative result does not change.

Consider a user whose utility function is ORA, that is given that he encountered one bad ad or more, it “erases” any positive experience from other ads in the same impression. A way to model it in the simulations is as follows. Let α, β denote the counts of relevant and irrelevant ads the user encountered, as before. The system accumulates the results of an impression, and increase α by the number of relevant ads the user clicked on if and only if he did not click on any bad ad. If he did click on at least one bad ad, only β increases, by one.

I experimented with the value of ranking by ORA, compared to ranking with LIN-LC, when the agents behave according to ORA. I got a consistent improvement of roughly 2.7% in total revenues after 2500 iterations (tradeoff knobs as above). However, it turns out that using ORA in the ranking model improves the long-term revenues even if the users behave according to LIN-LC, but by a smaller margin, 1.6% (after 2500 rounds).

In the last simulation, I modeled a user whose utility function is LCBG, that is the cost incurred by a bad ad is alleviated when another ad was helpful. Again I reflected the overall

experience through the relevance count, as follows. A user whose utility function is LCBG does not count the irrelevant ads within an impression that also produced relevant ads. Here the results were clearly positive: when users behave according to LCBG, the system that ranks with that function achieved about 2.8% higher total revenue (same parameters as above). When users behave according to LIN-LC, the two ranking systems (using LCBG and LIN-LC) achieved roughly the same amounts.

To conclude, I found a correlation between improvement in long-term revenues with a correct modeling of the additive and non-additive utility function. As an exception, it seems that using the more risk-averse ORA utility model in the ranking is generally beneficial in the long-term.

7.9 Summary

The revenue model of sponsored search relies on the user clicks, which in turn depend on whether or not the ads provide positive utility to the user. Therefore, to maintain search traffic and ad-revenue for the long-term, it is important to address the user's experience in the ad selection mechanism. For this reason, I studied methods for incorporating the user utility in Sponsored Search auctions. I proposed mechanisms in which utility user models can be incorporated, and studied their properties via simulations. I also simulated long-term effect of the new mechanism on the revenue, and showed that incorporating user's utility in the GSP mechanism, on the expense of short-term revenue, may increase the overall revenues in the long-term. Simulations also provided indications that correct modeling of user's utility can improve those long-term revenues.

Chapter 8

Conclusions

In this work I bring together two distinct areas of research: structured preference representation, and multiattribute auctions. An auction is in essence a (multi-agent) preference elicitation mechanism. In multiattribute auctions in particular, the preference elicitation problem is complicated due to the typically large domain and the difficulty of comparing outcomes over a large number of attributes. I showed how research results on structured preference representation can be leveraged to simplify representation and elicitation of preferences over such domains. My main contribution is therefore the novel application of modern preference-handling tools in electronic trading, where it is much needed but has not yet been applied systematically. My other contributions are in the area of structured preference representation, where the trading domain serves as an inspiration for my work on graphical models.

The first contribution to preference representation is in the extension of previous key results in this field, which originally rely on the vNM framework, to the trading domain (Chapter 3). The GAI framework, as defined by Fishburn (1967a), and later adapted to capture the structure resulting from a collection of the more intuitive CAI conditions (Bacchus and Grove, 1995), is limited in the previous treatments to preferences over lotteries. The willingness-to-pay representation does not involve lotteries, therefore the GAI factorization of this type of functions requires its own representation theorem.

My main contribution in the area of preference representation is the introduction of CUI networks (Chapter 6). (Conditional) Utility Independence is well-studied independence concept, which relies on a relatively intuitive notion: the invariance of preference order over one set of attributes to the value of another set of attributes. This invariance condition qualifies UI under any cardinal preferences model, in contrast to additive independence which rises from a fundamentally different set of axioms for each such model (see Section 2.1.4). Moreover, UI is a more general independence concept than additive independence, and not necessarily symmetric: it can be applied in one direction and allow the accommodation of a potential dependency in the other direction. The CUI network

is the first modern graphical model for utility independence. Due to the weaker independence concept, it is capable of simplifying more complex domains where the stakes are high enough to motivate the most accurate representation possible. This motivation points back to the multiattribute electronic trading domain, which I believe provides the appropriate challenge and motivation for this new representation. For this reason some of the algorithmic parts of the contribution are geared towards trading, and the main example of the chapter is based on a procurement problem. I stress, however, that CUI networks are as applicable to other domains, including typical AI problems involving decisions under uncertainty.

In the multiattribute auctions area, the main difference between my approach and previous literature is in the application of advanced preference handling tools. Although one might argue that multiattribute utility theory has been applied before in this domain through use of the additive representation, such treatment is seriously limiting. In most interesting cases preferences are not additive, and claims regarding efficiency of additive mechanisms are not accurate when preferences exhibit interdependencies. In fact, as I show experimentally in Chapter 4, auctions that are limited to an additive representation may lose significant portions of the potential trading surplus. The approximately efficient mechanism I introduce in Chapter 4 is unique compared to previous literature, by being both expressive (traders' preferences can exhibit interdependencies among attributes) and tractable (performance is exponential only in the size of the GAI elements).

Furthermore, in Chapter 5 I introduce a practically important domain in which the additive representation is clearly short of capturing crucial preferences. These preferences are typically referred to as allocation constraints. I show that despite the inherent dependency that these constraints impose on different trades, preferences are still amenable to meaningful factorization using the GAI representation. In Chapter 7 I introduce another domain in which the additive representation misses potentially useful preference information. Here, an additive representation of the search user's utility leads to desirable economic properties of the resulting mechanism, but fails to account for natural dependencies between the user's value from one advertisement and his experience from the other advertisements. I show how independence concepts introduced earlier can account for interdependencies but yet reduce the representation to single-dimensional functions. It is crucial to understand the utility of search users from sponsored-search advertisements, and incorporate it into the auction mechanism, in order to retain the tremendous economic benefits of this application.

Multiattribute mechanism design is interesting for additional reasons, beyond the multi-dimensional preference elicitation challenge. First, multiattribute mechanisms are typically applied in highly valued business settings. Therefore, the preference elicitation problem

gets even more difficult due to the strategic sensitivity of the preference information. As for any trading mechanism for selfish agents, we must carefully evaluate the strategic considerations of traders whose goal is to maximize their own utility. However, in this domain in particular we must also consider the future implications of information revelation by the traders. This privacy of information problem, which is in my view a major roadblock to the practical implementation of previous multiattribute auctions research, reflects another major distinction between my approach and most previous treatments of the domain. The buyer's preference information revealed to suppliers through the mechanisms proposed in Chapters 4 and 5 is limited to be a small fraction of the utility function, as I show experimentally in Chapter 4). As iterative auctions they also require limited revelation by the suppliers.

The multiattribute allocation problem exhibits a rich mathematical structure, which I believe can be leveraged to other areas in computer science, beyond trading. Combinatorial auctions have been shown to apply to various problems involving allocation of multiple resources to potentially selfish agents. Similarly, multiattribute auctions can capture various problems in which the allocation involves a *multi-objective decision*. That is, we need not only to determine to which agent a resource should be allocated, but we also have to consider how to configure the resource, while extracting preferences of the agents over the configurations. For example, consider a server, that can allocate portions of its incoming traffic to competing agents. The server can allocate sockets of different sizes, the larger ones being more expensive to the server. The agents have different valuations over the attention of the server, and in addition different requirements in terms of the volume of traffic they need to deliver, and they therefore need to negotiate on the volume attribute, in addition to price. A direction for future research in this niche is therefore taking multiattribute auctions closer to the core of computer science research.

Another potential direction is in other applications of structured preferences in electronic commerce. There are other social applications that could potentially benefit from preference factorization, such as recommendation, voting, and ranking systems. A third direction is to take a main idea developed here—the employment of advanced preference representations in automation problems—to other fields in AI which could benefit from it. As I argue in the introduction, preferences are a key component of most core AI areas. One approach, taken in Chapter 6, is to separate the preference representation problem, simplify it as much as possible, and apply it in whichever problem it is needed. It is plausible that in some areas utility factorization can be employed directly within the reasoning framework, as done here for multiattribute auctions. For example, Guestrin et al. (2003) apply preference decomposition to Markov decision processes.

To summarize, in this work I investigated the combination of preference handling and electronic trading. This combination promises significant benefits to electronic trading applications, which in turn inspire and yield advances of structured preference handling approaches. Beyond the accomplishments in this thesis, there are several promising directions in which this research can progress further.

Appendices

Appendix A

Proofs

A.1 Chapter 3

A.1.1 Theorem 3.1

Proof. As clear from the development leading to (3.2), given the CUI conditions the rhs of (3.2) is the same for any choice of pair Y^1, Y^2 . WLOG I assume $Y^1 \preceq Y^2$ and $X^0 \preceq X$. Now suppose that $g(\cdot) > 1$, meaning

$$u(X, Y^2, Z) - u(X, Y^1, Z) > u(X^0, Y^2, Z) - u(X^0, Y^1, Z),$$

or equivalently

$$u(X, Y^2, Z) - u(X^0, Y^1, Z) > u(X^0, Y^2, Z) - u(X^0, Y^1, Z) + u(X, Y^1, Z) - u(X^0, Y^1, Z).$$

This holds for any choice of X^0 . The reverse direction of (1) follows immediately, and (2) follows with opposite inequality signs. When we use an equal sign above, we get the definition of CDI. \square

A.1.2 Theorem 3.2

Proof. The proof is based on Keeney and Raiffa (1976) (KR), as explained below.

Assume that $u(\cdot)$ is normalized such that $u(A^0) = 0$. For each attribute $a \in A$, let $\bar{a} = \overline{\{a\}}$, and we know $UI(\bar{a}, a)$, therefore from Equation (2.1),

$$u(A) = f(a) + g(a)u(\bar{a}, a^0)$$

With the assignment \bar{a}^0 we get

$$u(\bar{a}^0, a) = f(a) + g(a)u(\bar{a}^0, a^0) = f(a)$$

Hence $f(a) = u(\bar{a}^0, a)$, and $g(a) = \frac{u(A) - u(\bar{a}^0, a)}{u(\bar{a}, a^0)}$ (this development is done by KR). With $u(A^0) = 0$, we get $g(a) = \frac{u(A) - u(\bar{a}^0, a)}{u(\bar{a}, a^0) - u(\bar{a}^0, a^0)}$. This is a special case of $g(\cdot)$ as defined above for the CUI condition. We can therefore use Theorem 3.1.

In proof of Theorem 6.1, KR define the MUI-factor as follows:

$$k = \frac{g(a) - 1}{u(\bar{a}^0, a)}$$

The denominator is always positive. From this and Theorem 3.1 we get the desired result. \square

A.1.3 Lemma 3.3 and Proposition 3.4

Proof. I prove Proposition 3.4. Lemma 3.3 falls out of the proof construction.

Let X^0, Y^0 be arbitrary instantiations. I define

$$\begin{aligned}\psi_1(X, Z) &= u(X, Y^0, Z) - u(X^0, Y^0, Z) \\ \psi_2(Y, Z) &= u(X^0, Y, Z)\end{aligned}$$

Note that ψ_1 does not depend on the choice of Y^0 due to the CDI condition,

$$\psi_1(X, Z) = u(X, Y, Z) - u(X^0, Y, Z).$$

Thus,

$$\begin{aligned}\psi_1(X, Z) + \psi_2(Y, Z) &= u(X, Y, Z) - u(X^0, Y, Z) + u(X^0, Y, Z) \\ &= u(X, Y, Z).\end{aligned}$$

That proves CDI sufficiency for the functional decomposition. For the other direction as-

sume there exist ψ_1 and ψ_2 as in (3.3) and decompose $u(A)$ as follows:

$$\begin{aligned} u(X, Y, Z) - u(X^0, Y, Z) \\ &= \psi_1(X, Z) + \psi_2(Y, Z) - (\psi_1(X^0, Z) + \psi_2(Y, Z)) \\ &= \psi_1(X, Z) - \psi_1(X^0, Z). \end{aligned}$$

Since the difference measurement does not depend on Y , we have $\text{CDI}(X, Y \mid Z)$. \square

A.1.4 Theorem 3.6

I first prove a somewhat weaker result.

Claim 1. *There exists a collection $\mathcal{Q} = \{C_1, \dots, C_w\}$ of cliques of G , that includes all the maximal cliques, and*

$$u(A) = \sum_{k=1}^w (-1)^{k+1} \sum_{1 \leq i_1 < \dots < i_k \leq w} u\left(\left[\bigcap_{s=1}^k C_{i_s}\right]\right). \quad (\text{A.0})$$

Proof. Let $G^0 = (A, E^0)$ be the complete graph over the nodes of G . Note that each edge $(x, y) \in E^0 \setminus E$ implies $\text{CDI}(x, y)$. I use induction on a series of edge removals. starting from the graph G^0 , at each step i we remove an edge in $E^0 \setminus E$ to get graph G^i . After the last step $i = |E^0| - |E|$ and $G^{|E^0| - |E|} = G$. The induction properties maintained after each step i are as follows:

1. Equation (1) holds for some collection $\mathcal{Q}^i = \{\hat{C}_1, \dots, \hat{C}_{w^i}\}$ of cliques of G^i .
2. For each maximal clique M of G there exists $\hat{C}_l \in \mathcal{Q}^i$ such that $M \subseteq \hat{C}_l$.

Since A is a clique in G^0 , in step 0, $\mathcal{Q}^0 = \{A\}$ and both properties trivially hold. Following the process for step 1 provides intuition as for how the final decomposition is obtained. We pick a pair of nodes (x, y) such that $\text{CDI}(x, y)$. By Lemma 3.3 (I use the notation $S^{-d} = S \setminus \{d\}$ for any set S):

$$\begin{aligned} u(A) &= u(x, y, A^{-x, y}) \\ &= u(x^0, y, A^{-x, y}) + u(x, y^0, A^{-x, y}) - u(x^0, y^0, A^{-x, y}) \\ &= u([A^{-x}]) + u([A^{-y}]) - u([A^{-x} \cap A^{-y}]). \end{aligned} \quad (\text{A.1})$$

Define $\mathcal{Q}^1 = \{A^{-x}, A^{-y}\}$. Equation (A.1) shows that (1) holds for \mathcal{Q}^1 , and it is easy to see that the second induction property holds for \mathcal{Q}^1 as well.

Assuming the properties 1 and 2 hold at step i , I will prove for step $i + 1$. Let (x, y) denote the edge removed in step $i + 1$. Let $\hat{C}_1, \dots, \hat{C}_d$ (WLOG) indicate all the sets in \mathcal{Q}^i

that include both x and y . Similar to (A.1), we observe that

$$u([\hat{C}_1]) = u([\hat{C}_1^{-x}]) + u([\hat{C}_1^{-y}]) - u([\hat{C}_1^{-x} \cap \hat{C}_1^{-y}]). \quad (\text{A.2})$$

Similarly for any $k = 1, \dots, w^i - 1$, and $1 < i_1 < \dots < i_k \leq w^i$,

$$u([\bigcap_{s=1}^k \hat{C}_{i_s} \cap \hat{C}_1]) = u([\bigcap_{s=1}^k \hat{C}_{i_s} \cap \hat{C}_1^{-x}]) + u([\bigcap_{s=1}^k \hat{C}_{i_s} \cap \hat{C}_1^{-y}]) - u([\bigcap_{s=1}^k \hat{C}_{i_s} \cap \hat{C}_1^{-x} \cap \hat{C}_1^{-y}]). \quad (\text{A.3})$$

In (1) (assumed to hold before this step) each term that includes \hat{C}_1 can be substituted according to (A.2) or (A.3). Doing so will result in (1) holding for the set $(\mathcal{Q}_i \setminus \{\hat{C}_1\}) \cup \{\hat{C}_1^{-x}, \hat{C}_1^{-y}\}$.

We repeat the same operation for C_2, \dots, C_d , and define the resulting collection

$$\mathcal{Q}^{i+1} = (\mathcal{Q}^i \setminus \{\hat{C}_1, \dots, \hat{C}_d\}) \cup \{\hat{C}_1^{-x}, \hat{C}_1^{-y}, \dots, \hat{C}_d^{-x}, \hat{C}_d^{-y}\}.$$

All elements in \mathcal{Q}^{i+1} are subsets of elements in \mathcal{Q}^i . Since the only difference between G^i and G^{i+1} is the removed edge (x, y) , and since no set in \mathcal{Q}^{i+1} includes both x and y , every set in \mathcal{Q}^{i+1} must be a clique in G^{i+1} . Also, if M is a maximal clique in G , then $M \subseteq \hat{C}_l$ for some $\hat{C}_l \in \mathcal{Q}^i$. Since M cannot include both x and y , If $l \in \{1, \dots, d\}$ then either $M \subseteq \hat{C}_l^{-x}$ and/or $M \subseteq \hat{C}_l^{-y}$. In any case M is contained within an element of \mathcal{Q}^{i+1} . This proves the induction step.

As a result, in the last step the decomposition (1) holds for the set $\mathcal{Q} = \mathcal{Q}^{|E_0| - |E|}$, which is a set of cliques in G . Induction property 2 ensures that $\mathcal{Q}^{|E_0| - |E|}$ includes all maximal cliques of G . \square

(Theorem 3.6). The result of Claim 1 can be strengthened such that the decomposition is restricted to maximal cliques. Though the resulting set \mathcal{Q} may include additional (smaller) cliques, or multiple elements for the same clique, all the terms that correspond to those redundant elements cancel themselves out. To see this, take an element $\hat{C}_p \in \mathcal{Q}$ which is not a max clique, or a duplicate element for a maximal clique. In both cases there exists (at least one) max clique $M = C_q \in \mathcal{Q}$ such that $C_p \subseteq C_q$. Clearly,

$$u([C_p]) = u([C_p \cap C_q]).$$

Since these two terms appear in (1) with opposite signs they cancel out. More generally, we assume WLOG that $q = 1$ and $p = r$. Then for any $k = 1, \dots, r^i - 2$, and $1 < i_1 < \dots < i_k < r$,

$$u([\bigcap_{s=1}^k C_{i_s} \cap C_p]) = u([\bigcap_{s=1}^k C_{i_s} \cap C_p \cap C_q]) \quad (\text{A.4})$$

Since the intersection on the right hand side is over one more element than the one on the left, their corresponding terms have opposite signs and cancel out. Clearly, any term in (1) must appear in on of the sides of an instance of (A.4). Therefore all these terms cancel out and we get (still $p = w$)

$$\sum_{k=1}^w (-1)^{k+1} \sum_{1 \leq i_1 < \dots < i_k \leq w} u([\bigcap_{s=1}^k C_{i_s}]) = \sum_{k=1}^{w-1} (-1)^{k+1} \sum_{1 \leq i_1 < \dots < i_k \leq w-1} u([\bigcap_{s=1}^k C_{i_s}]).$$

Meaning that (1) holds for the collection of cliques $\mathcal{Q} \setminus \{C_p\}$. We do the same for the next non maximal clique or duplicate element in $\mathcal{Q} \setminus \{C_p\}$ and ultimately get that (1) holds for the set of g maximal cliques of G .

From here, it is easy to see that the functions f_1, \dots, f_g can be defined according to (3.5). \square

A.1.5 Theorem 3.7

Proof. To qualify as an MVF, we first need to define an order over preference differences.

Definition A.1. Let \preceq be a quasi-linear preference order over (A, p) , and let $\theta_1, \theta_2, \theta_3, \theta_4 \in \Theta$. Then

$$[\theta_1, \theta_2] \preceq^* [\theta_3, \theta_4]$$

iff for any values p^1, p^2, p^3, p^4 of p such that $(\theta_1, p^1) \sim (\theta_2, p^2)$ and $(\theta_3, p^3) \sim (\theta_4, p^4)$,

$$p^1 - p^2 \leq p^3 - p^4. \quad (\text{A.5})$$

Due to quasi-linearity, if Definition A.1 holds for one set of prices, it holds for any (under the indifference restriction). It is immediate from the definition that the WTP function is an MVF over \preceq^* . The interpretation of the statement $[\theta_1, \theta_2] \preceq^* [\theta_3, \theta_4]$ is that “willingness to pay to improve from θ_2 to θ_1 is no greater than willingness to pay to improve from θ_4 to θ_3 ”. \square

A.2 Chapter 4

A.2.1 Proofs of Section 4.3.1

Lemma 4.1

Proof. Let θ be a consistent cover in Φ . Let $\mu \subset \{1, \dots, g\}$ be any subtree of the GAI tree. Let γ be the parent of the root of μ , or an arbitrary element out of μ if μ is disconnected from the rest of the graph. Let $\hat{\theta} \in \Psi$ be the source of θ_γ . Let y be the number of elements in μ , and finally let $\theta_{\mu_1}, \dots, \theta_{\mu_y}$ denote the sub-configurations in θ corresponding to the elements in μ . I define the operation $trim(\mu, \theta)$ as follows:

$$trim(\mu, \theta) = (\hat{\theta}_{\mu_1}, \dots, \hat{\theta}_{\mu_y}, \theta_{\bar{\mu}_1}, \dots, \theta_{\bar{\mu}_{g-y}}).$$

That is we replace each of θ_{μ_r} by the corresponding sub-configuration in $\hat{\theta}$. $\theta' = trim(\mu, \theta)$ must be consistent: the elements $\hat{\theta}_{\mu_1}, \dots, \hat{\theta}_{\mu_y}$ are internally consistent because they have a mutual source $\hat{\theta}$, the other elements are internally consistent because they are both sub-configurations of θ , and these two subsets can only intersect between the root of μ and γ , which are consistent having a mutual source $\hat{\theta}$. I call a trim operation *simple* if $\theta_{\mu_1}, \dots, \theta_{\mu_y}$ have a mutual source. A trivial trim is a trim of a disconnected component μ .

Claim 2. *Let Ψ and Φ be defined as above, and let θ denote a suboptimal consistent cover in Φ . If $\theta' = trim(\mu, \theta)$, and the trim operation is simple, then θ' must also be suboptimal.*

Proof. Let $\tilde{\theta} \in \Psi$ denote the single source of $\theta_{\mu_1}, \dots, \theta_{\mu_y}$. If θ' is optimal, meaning $\pi_\tau(\theta') > \pi_\tau(\theta)$, then

$$\pi_\tau(\theta') = \pi_\tau(\theta'_\mu) + \pi_\tau(\theta'_{\bar{\mu}}) > \pi_\tau(\theta_\mu) + \pi_\tau(\theta_{\bar{\mu}}).$$

Since $\theta_\mu = \theta'_\mu$

$$\pi_\tau(\theta'_\mu) > \pi_\tau(\theta_\mu)$$

Define the following cover:

$$\hat{\theta} = (\theta'_{\mu_1}, \dots, \theta'_{\mu_y}, \tilde{\theta}_{\bar{\mu}_1}, \dots, \tilde{\theta}_{\bar{\mu}_{r-y}})$$

$\hat{\theta}$ is consistent, since again the only intersection is between the root of μ (in θ') which is consistent with the same element in θ , and its parent (in $\tilde{\theta}$) which is also consistent with

its parallel in θ . Now

$$\pi_\tau(\hat{\theta}) = \pi_\tau(\theta'_\mu) + \pi_\tau(\tilde{\theta}_\mu) > \pi_\tau(\theta_\mu) + \pi_\tau(\tilde{\theta}_\mu) = \pi_\tau(\tilde{\theta}).$$

The last equation follows from the fact that all sub-configurations of θ_μ are from $\tilde{\theta}$. This contradicts optimality of $\tilde{\theta} \in \Psi$. \square

Let θ^1 be a consistent cover of sub-configurations from configurations in Ψ such that $\pi_\tau(\theta^1)$ is suboptimal. We now perform a series of *trim* operations, resulting in a series $\theta^1, \dots, \theta^L$ ($L \leq g$). The last element θ^L is such that all its element have a mutual source, meaning $\theta^L \in \Psi$. If I show that in each step the trim is *simple*, the result θ^g must be suboptimal contradicting optimality of Ψ .

I define a series μ_1, \dots, μ_g , by traversing each tree in order corresponding to backtracking in Depth-First-Search: that is, starting from the rightmost leaf, next its siblings, next their parent, and in general once all its children of a node γ_j are visited, the next μ_j is the subtree of γ_j . Each time we get to a leaf we start from an empty set again, therefore each μ_j corresponds to a subtree. Clearly, $\text{trim}(\mu_j, \theta_j)$ is simple because for each leaf $|\mu_j| = 1$, and after $\text{trim}(\mu_j, \theta_j)$ all elements of μ_{j+1} have a mutual source therefore $\text{trim}(\mu_{j+1}, \theta_{j+1})$ is also simple. It is possible that $L < g$ because if the subtree μ_j and its parent γ already have a mutual source no trim is needed. \square

Lemma 4.3

Proof. We perform the same series of *trim* operations as in the proof of Lemma 4.1. However, here $\pi_\tau^t(\theta^{j+1})$ can be up to δ above configurations in Ψ without breaking δ -optimality of Ψ . Therefore instead of claiming that if θ^j is suboptimal so is θ^{j+1} , I claim $\pi_\tau^t(\theta^{j+1}) \leq \pi_\tau^t(\theta^j) + \varepsilon$. The number of trim operations is clearly bounded by $g - 1$, therefore after the final trim L , $\pi_\tau(\theta^L) \leq \pi_\tau(\theta^1) + (g - 1)\delta$. Let $M = \max_{\theta \in \Theta} \pi_\tau(\theta)$. If $\pi_\tau(\theta^1) < M - g\delta$ then $\pi_\tau(\theta^L) < M - \delta$ and this is a contradiction since $\theta^L \in \Psi$. \square

Lemma 4.5

Proof. I prove the lemma per each tree G_j . The optimal values for disconnected components are independent of each other hence if the maximal profit for each component does not change the combined maximal profit does not change as well. If the price of θ'_j was

reduced during phase A, that is $p^{t+1}(\theta'_j) = p^t(\theta'_j) - \delta$, it must be the case that some $w \leq g_j$ sub-configurations of θ'_j are not in \mathcal{M}_j^t , and $\delta = \frac{w\varepsilon}{g}$. The definition of \mathcal{M}_j^t ensures

$$\pi_b^t(\theta'_j) < \max_{\theta \in \Theta} \pi_b^t(\theta_j) - g_j \frac{\varepsilon}{g}.$$

Therefore,

$$\pi_b^{t+1}(\theta') = \pi^t(\theta') + \delta = \pi^t(\theta') + \frac{w\varepsilon}{g} \leq \max_{\theta \in \Theta} \pi_b^t(\theta_j).$$

This is true for any configuration whose profit improves, therefore the maximal buyer profit does not change during phase A. \square

Lemma 4.6

Proof. In each round $t < T$ of phase A there exists an active seller i for whom $B_i^t \cap M^t = \emptyset$. However to be active in round t , $B_i^t \neq \emptyset$. Let $\hat{\theta} \in B_i^t$. If $\forall r. \hat{\theta}_r \in \mathcal{M}^t$, then $\hat{\theta} \in M^t$ by definition of M^t . Therefore there must be $\hat{\theta}_r \notin \mathcal{M}^t$. \square

Lemma 4.7

Proof. By Lemma 4.6 prices must go down in every round of phase A. Lemma 4.5 ensures a lower bound on how much prices can be reduced during phase A, therefore the auction either terminates in phase A or must reach condition [SWITCH].

We set the initial prices are high such that $\max_{\theta \in \Theta} \pi_b^1(\theta) < 0$, and by Lemma 4.5 $\max_{\theta \in \Theta} \pi_b^t(\theta) < 0$ during phase A. The assumption is that the efficient allocation (θ^*, i^*) provides positive welfare, that is $\sigma_{i^*}(\theta^*) = \pi_b^t(\theta^*) + \pi_{i^*}^t(\theta^*) > 0$. s_{i^*} is SB therefore she will leave the auction only when $\pi_{i^*}^t(\theta^*) < 0$. This can happen only when $\pi_b^t(\theta^*) > 0$, therefore s_{i^*} does not drop in phase A hence the auction cannot terminate before reaching condition [SWITCH]. \square

Lemma 4.8

Proof. η_i is chosen to maximize the buyer's surplus out of B_i^t at the end of phase A. Since $B_i^t \cap M^t \neq \emptyset$, clearly $\eta_i \in M^t$. From Corollary 4.4 and Corollary 4.2, for any $\tilde{\theta}$,

$$\begin{aligned}\pi_b^T(\eta_i) &\geq \pi_b^T(\tilde{\theta}) - (e+1)\varepsilon \\ \pi_i^T(\eta_i) &\geq \pi_i^T(\tilde{\theta}) \\ \Rightarrow \sigma_i(\eta_i) &\geq \sigma_i(\tilde{\theta}) - (e+1)\varepsilon\end{aligned}$$

□

Lemma 4.9

Proof. From SB and the definition of phase B, $s_{\tilde{i}}$ drops when $\Delta > \pi_{\tilde{i}}^T(\eta_{\tilde{i}})$. If s_i did not drop before that point then $\pi_i^T(\eta_i) \geq \Delta - \varepsilon > \pi_{\tilde{i}}^T(\eta_{\tilde{i}}) - \varepsilon$. Together with $\eta_i \in M^t$ we get

$$\pi_b^T(\eta_i) + \pi_i^T(\eta_i) \geq \max_{\theta \in \Theta} \pi_b^T(\theta) + \pi_{\tilde{i}}^T(\eta_{\tilde{i}}) - (e+2)\varepsilon.$$

From Corollary 4.2, $\pi_{\tilde{i}}^T(\eta_{\tilde{i}}) = \max_{\theta \in \Theta} \pi_{\tilde{i}}^T(\theta)$. Therefore

$$\begin{aligned}\sigma_i(\eta_i) &= \pi_b^T(\eta_i) + \pi_i^T(\eta_i) \geq \\ &\max_{\theta \in \Theta} \pi_b^T(\theta) + \max_{\theta \in \Theta} \pi_{\tilde{i}}^T(\theta) - (e+2)\varepsilon \geq \max_{\theta \in \Theta} \sigma_{\tilde{i}}(\theta) - (e+2)\varepsilon.\end{aligned}$$

□

Theorem 4.10

Proof. From Lemma 4.7 the auction terminates with an allocation (s_i, η_i) . From Lemma 4.8, the theorem is immediate in case the winning seller s_i is the efficient seller. Otherwise the efficient seller is $s_{\tilde{i}}$ who dropped before or with s_i . The result is now immediate from Lemma 4.9. □

Lemma 4.11

Proof. Trivially, we consider only the winning seller s_i . First, assume the final price is not above the buyer's valuation. The payment to the winning seller is $p^T(\eta_i) - \Delta$. Let s_j be the

second best seller. From the timing that s_j drops,

$$\Delta + \varepsilon > \pi_j^T(\eta_j) = \max_{\theta \in \Theta} \pi_j^T(\theta). \quad (\text{A.6})$$

From Corollary 4.4,

$$u_b(\eta_i) - p^T(\eta_i) \geq \max_{\theta \in \Theta} \pi_b^T(\theta) - (e+1)\varepsilon.$$

Therefore (using (A.6))

$$\begin{aligned} p^T(\eta_i) - \Delta &\leq u_b(\eta_i) - \max_{\theta \in \Theta} \pi_b(\theta) + (e+1)\varepsilon - \Delta < \\ u_b(\eta_i) - \max_{\theta \in \Theta} \pi_b^T(\theta) + (e+2)\varepsilon - \max_{\theta \in \Theta} \pi_j^T(\theta) &\leq u_b(\eta_i) - \max_{\theta \in \Theta} \sigma_j(\theta) + (e+2)\varepsilon \end{aligned} \quad (\text{A.7})$$

Also from the timing of s_j 's drop

$$\Delta \leq \pi_j^T(\eta_j) + \varepsilon$$

Meaning:

$$p^T(\eta_j) - \Delta \geq c_j(\eta_j) - \varepsilon \quad (\text{A.8})$$

From Corollary 4.4

$$u_b(\eta_i) - p^T(\eta_i) \leq u_b(\eta_j) - p^T(\eta_j) + (e+1)\varepsilon$$

Therefore (using (A.8))

$$\begin{aligned} p^T(\eta_i) - \Delta &\geq u_b(\eta_i) - u_b(\eta_j) + p^T(\eta_j) - (e+1)\varepsilon - \Delta \geq \\ u_b(\eta_i) - (u_b(\eta_j) - c_j(\eta_j)) - \varepsilon - (e+1)\varepsilon &\geq u_b(\eta_i) - \max_{\theta \in \Theta} \sigma_j(\theta) - (e+2)\varepsilon \end{aligned} \quad (\text{A.9})$$

Equations (A.7) and (A.9) place the payment $p^T(\eta_i) - \Delta$ within $(e+2)\varepsilon$ from s_i 's VCG payment.

In the case that final price is above buyer's valuation the payment $u_b(\eta_i)$ is exactly the VCG payment. \square

Theorem 4.12

Proof. Let s_1 play some arbitrary strategy ρ_1 against SB sellers s_2, \dots, s_n . If s_1 does not win she would clearly have done no worse using SB, therefore we assume s_1 wins η_1 in final price \tilde{p} and that she gains at least $(3e+5)\varepsilon$ from the trade. Let $i \in 2, \dots, n$. The calculation of (A.7) assumed nothing on the winning trader's strategy, therefore it applies here as well:

$$\tilde{p} = p^T(\eta_1) - \Delta \leq u_b(\eta_1) - \max_{\theta} \sigma_i(\theta) + (e+2)\varepsilon \quad (\text{A.10})$$

Next, define the following cost function: $\hat{c}_1(\eta_1) = \tilde{p} - (2e+3)\varepsilon$ and $\hat{c}_1(\theta') = \infty, \forall \theta' \neq \eta_1$. Assume s_1 plays SB for \hat{c}_1 .

Claim 3. *By playing SB assuming cost \hat{c}_1 , s_1 is still the winner, and her profit (wrt to $c_1(\cdot)$) is within $(2e+3)\varepsilon$ of her profit playing ρ_1 .*

Proof. Clearly, s_1 bids only on η_1 . Let $\hat{p}(\cdot)$ denote prices in the end of phase A in the new instance of the auction, let $\hat{\pi}_b(\cdot)$ denote the buyer's profit, and let $\hat{\Delta}$ be the final discount. Now assume that prices reach s_1 's limit, that is $\hat{\Delta} = \hat{p}(\eta_1) - (\tilde{p} - (2e+3)\varepsilon)$.

Using (A.10),

$$\begin{aligned} \hat{\Delta} &= \hat{p}^T(\eta_1) - \tilde{p} + (2e+3)\varepsilon \\ &> u_b(\eta_1) - \hat{\pi}_b^T(\eta_1) - (u_b(\eta_1) - \max_{\theta} \sigma_i(\theta) + (e+2)\varepsilon) + (2e+3)\varepsilon \\ &= \max_{\theta} \sigma_i(\theta) - \hat{\pi}_b^T(\eta_1) + (e+1)\varepsilon \end{aligned} \quad (\text{A.11})$$

Let $\hat{\eta}_i$ denote the configuration chosen for i at the end of phase A in the new instance. Since $\hat{\eta}_i \in M^T$ in the second instance, we get that $\hat{\pi}_b^T(\hat{\eta}_i) \geq \hat{\pi}_b^T(\eta_1) - (e+1)\varepsilon$. Therefore

$$\hat{\Delta} \geq \sigma_i(\hat{\eta}_i) - \hat{\pi}_b^T(\hat{\eta}_i) = \hat{p}^T(\hat{\eta}_i) - c_i(\hat{\eta}_i) \quad (\text{A.12})$$

That shows that s_1 wins in the second instance as well, and since the value of $\hat{\Delta}$ above is the maximal value in which the best $s_i, i = 2, \dots, n$ leaves the auction, the final price \hat{p} paid to s_1 is at least $\tilde{p} - (2e+3)\varepsilon$. \square

From Lemma 4.11:

$$\hat{p} \leq VCG(\hat{c}_1, c_2, \dots, c_n) + (e+2)\varepsilon$$

Truthful reporting is a dominant strategy for sellers in one-sided VCG auctions. Therefore

$$VCG(\hat{c}_1, c_2, \dots, c_n) \leq VCG(c_1, c_2, \dots, c_n).$$

With the result of the claim we get

$$\tilde{p} \leq \hat{p} + (2e + 3)\varepsilon \leq VCG(c_1, c_2, \dots, c_n) + (3e + 5)\varepsilon.$$

Therefore by playing ρ_1 , s_1 could not have gained more than $(3e + 5)\varepsilon$ above her worst-case payoff for playing SB with respect to her true cost c_1 . \square \square

A.2.2 Proposition 4.13

Proof. For simplicity of notations I assume that the GAI-tree is connected. The extension to multiple connected components is immediate since each component is optimized separately.

As mentioned, functions in GAI format can be optimized using variable elimination schemes. In particular I refer to Nilsson (1998), on which the partitioning method below is based.

Finding the best configuration is exponential in the tree-width of the CDI-map and therefore linear in $|I|$. The sub-configurations of the optimum is obviously in \mathcal{M}^t . Next we need to find the second-best configuration, and so on. Each such optimization is equivalent to finding the optimal configuration in $\Theta \setminus \mathcal{M}^t$, and this can be done by first performing each of the following g optimizations:

$$\Theta_i = \{\theta \in \Theta \mid \theta_1, \dots, \theta_{i-1} \in \mathcal{M}^t \text{ and } \theta_i \notin \mathcal{M}^t\}$$

$$\theta^i = \max_{\theta \in \Theta_i} \pi_b^t(\theta)$$

.

The best configuration in $\Theta \setminus \mathcal{M}^t$ will simply be

$$\theta^* = \max_{i=1, \dots, g} \pi_b^t(\theta^i)$$

If $\pi_b^t(\theta^*) \geq \max_{\theta \in \Theta} \pi_b^t(\theta) - \varepsilon$, then each sub-configuration of θ^* that is not already in \mathcal{M}^t is added to \mathcal{M}^t . Otherwise, \mathcal{M}^t is ready.

Each maximization adds at least one sub-configuration to \mathcal{M}^t or ends the computation for round t . Therefore the number of times this task is done per round is bounded by

the number of sub-configurations $|I|$ plus one. Moreover, \mathcal{M}^t is monotonically increasing in the auction hence the total number of times this task is done throughout the auction is bounded by $|I| + T$. By considering that each such task takes time $O(|I|)$ we get the required results. \square

A.3 Chapter 5

A.3.1 Theorem 5.1

Proof. By linear trade utility,

$$u_b([t_j]) = g_b(q_j)u_b([t_j^1])$$

Therefore, for any $T \subset \mathcal{T}$, based on the GAI decomposition in (5.3):

$$\begin{aligned} \sum_{b \in B} u_b(T_b) - \sum_{s \in S} u_s(T_s) = & \sum_{b \in B} \sum_{j=1}^{|S|} g_b(q_j)u_b([t_j^1]) - \sum_{s \in S} \sum_{j=1}^{|B|} g_s(q_j)u_s([t_j^1]) + \sum_{b \in B} (u_b([\vec{q}_b]) - \\ & \sum_{j=1}^{|S|} u_b([q_j])) - \sum_{s \in S} (u_s([\vec{q}^s]) - \sum_{j=1}^{|B|} u_s([q_j])) \quad (\text{A.13}) \end{aligned}$$

Let $t' = (\theta', q', b', s')$, such that $q' < \underline{q}_{b'}$. Clearly, under free disposal (and our non-zero utility assumption) a solution including such trade cannot be optimal, since the total welfare is higher without that trade. A trade for which $q' > \bar{q}_{s'}$ will not occur due to infinite cost. If $q' > \bar{q}_{b'}$, modifying the trade with $q' = \bar{q}_{b'}$ cannot decrease welfare: it does not decrease b' 's utility and cannot decrease a seller's cost (recall that the claim is that *there exists* a trade that consists of solutions to *MMP*). We can therefore assume that $\forall b, j, q_j$ is within both ranges, and $g_b(q_j) = g_{s_j}(q_j) = q_j$.

We can now replace $g_b(q_j)$ and $g_{s_j}(q_j)$ in (A.13) with q_j . Using (5.1) we observe,

$$\begin{aligned} \sum_{b \in B} \sum_{j=1}^{|S|} q_j u_b([t_j^1]) - \sum_{s \in S} \sum_{j=1}^{|B|} q_j u_s([t_j^1]) = & \sum_{b \in B} \sum_{j=1}^{|S|} q_j (u_b([t_j^1]) - u_{s_j}([t_j^1])) = \sum_{b \in B} \sum_{j=1}^{|S|} q_j (\mu_b(\theta_j, s_j) - \mu_{s_j}(\theta_j, b)) \quad (\text{A.14}) \end{aligned}$$

From (A.13) and (A.14), the solution to GMAP is equivalent to

$$\max_{T \subset \mathcal{T}} \left(\sum_{b \in B} \sum_{j=1}^{|S|} q_j (\mu_b(\theta_j, s_j) - \mu_{s_j}(\theta_j, b)) + \sum_{b \in B} (u_b([\vec{q}_b]) - \sum_{j=1}^{|S|} u_b([q_j])) - \left(\sum_{s \in S} (u_s([\vec{q}^s]) - \sum_{j=1}^{|B|} u_s([q_j])) \right) \right) \quad (\text{A.15})$$

For any fixed choice of \vec{q} , this term is maximized by picking the maximal value for each of the terms $(\mu_b(\theta_j, s_j) - \mu_{s_j}(\theta_j, b))$, each of which is a solution to MMP for the pair of traders of trade t_j . Note that the values for \vec{q} are chosen independently of configurations.

To prove the necessity of configuration aggregation, we assume that it does not hold, that is there exists a trader b_1 whose utility u_{b_1} includes a GAI element I_d that includes either the pair of attributes (a_j^i, a_k^h) for some attribute indices i, h and trades $j \neq k$, or the pair (a_j^i, q_k) (again $j \neq k$). It is sufficient to show an instance of the problem in which the solution to GMAP includes suboptimal solutions to MMP. Indeed, assume there exists a utility component $f(a_j^i, a_k^h)$ (potentially depending on more attributes). Let $\hat{\theta} = \hat{a}_j^1, \dots, \hat{a}_j^m$ be a bilaterally suboptimal configuration to b_1 and s_k , and $\tilde{\theta} = \tilde{a}_k^1, \dots, \tilde{a}_k^m$ an optimal configuration (solves $\text{MMP}(b_1, s_k)$).

We could set $f(\hat{a}_j^i, \tilde{a}_k^h)$ high enough to cause GMAP to pick $\hat{\theta}$ for the trade between b_1 and s_j . Similarly this can be done in the case that the additional component is (a_j^i, q_k) . \square

A.3.2 Theorem 5.2

Proof. Let \hat{T} represent the set of trades which is the solution to GMAP. Define $n = |S|$, and let q'_1, \dots, q'_n represent the quantity allocated to each seller (some might be zero), for the configurations $\theta_1, \dots, \theta_n$ respectively. We decompose buyer's utility by (5.5). Then

$$\sigma(\hat{T}) = \sum_{j=1}^n (q'_j (\mu_b(\theta_j) - \mu_j(\theta_j)) + f_b^q(\vec{q})) \quad (\text{A.16})$$

By Lemma 4.8, $\forall j = 1, \dots, n$,

$$\mu_b(\theta_j) - \mu_j(\theta_j) \leq \varepsilon(e+1) + \mu_b(\eta_j) - \mu_j(\eta_j)$$

And since $q = \sum_{i=1}^n q'_i$,

$$\sum_{j=1}^n (q'_j(\mu_b(\theta_j) - \mu_j(\theta_j))) \leq q\epsilon(e + 1) + \sum_{j=1}^n (q'_j(\mu_b(\eta_j) - \mu_j(\eta_j))) \quad (\text{A.17})$$

Note that the *winner determination problem (WDP)* solved by Phase B is almost exactly *GMAP*, with the one difference that each seller s_j supplies the configuration η_j . We can construct a solution to the *WDP* as follows: take the solution to *GMAP* from (A.16), and replace each θ_j with η_j . This is a valid solution to Phase B *WDP*, and yet (by (A.17)) it must be within $q\epsilon(e + 1)$ from the solution to *GMAP*. \square

A.3.3 Theorem 5.3

Proof. We assume all traders play SB, and examine how much can s_1 gain by deviating.

Phase B is a (seller-side) VCG mechanism, whose only inputs are the values of $v_i = \pi_i(\eta_i) \forall i$ (for the purpose of the proof we can assume that sellers that dropped out during Phase A, continued to bid with negative valuations, and participate in Phase B with their appropriate negative values for v_i).

Therefore, the only way in which s_1 could gain is by affecting the values of v_i . To improve her own profit, s_1 would either improve her own value v_1 or reduce her competitors' values. Note that $\sigma_i(\eta_i) = \pi_b(\eta_i) + v_i$. By Lemma 4.8, if s_i plays SB,

$$MMP_i \geq \pi_b(\eta_i) + v_i \geq MMP_i - \epsilon(e + 1)$$

From Lemma 4.5 the buyer's maximal utility during Phase A is a constant MU , and therefore

$$MU \geq \pi_b(\eta_i) \geq MU - \epsilon(e + 1)$$

Together it shows that s_1 cannot affect v_i by more than $2\epsilon(e + 1)$. Note that this potential gain is per unit, therefore totals in $2q\epsilon(e + 1)$

The same two equations hold for v_1 if s_1 was to play SB, therefore by deviation s_1 cannot improve v_1 by more than $2\epsilon(e + 1)$ compared to playing SB. This is again per unit, therefore the total possible gain from deviation is $4q\epsilon(e + 1)$. \square

A.4 Chapter 6

A.4.1 Proposition 6.1

Proof. By the definitions of $Pa(x)$ and $P(x)$, (6.3) holds when replacing $Dn(x)$ with $\tilde{Dn}(x)$. From the definition of CUI, it is straightforward that

$$\text{CUI}(A \setminus (Y \cup W), Y \mid W) \Rightarrow \text{CUI}(A \setminus (Y \cup W \cup Z), Y \mid W \cup Z),$$

because invariance of preference order over $A \setminus (Y \cup W)$ implies invariance of preference order over its subset $A \setminus (Y \cup W \cup Z)$, when the difference set Z is fixed. Given (6.2), and taking $W = Pa(x) \cup \tilde{Dn}(x)$ and $Z = Dn(x) \setminus \tilde{Dn}(x)$, we get (6.3). \square

A.4.2 Lemma 6.3

Proof. Let $Z = A \setminus (X \cup Y)$ and $C = A \setminus (A \cup B)$. We simply apply the two independence conditions consequentially, and we can define \hat{f}, \hat{g} such that:

$$\begin{aligned} u(A) = u(XYZ) &= f(YZ) + g(YZ)u([A \setminus Y]) = f(YZ) + g(YZ)(f'((BC) \setminus Y) \\ &\quad + g'((BC) \setminus Y)u([A \setminus (YB)])) = \hat{f}(ZBYC) + \hat{g}(ZBYC)u(A \setminus (YB)). \end{aligned}$$

Since $Z \cup Y \cup B \cup C = A \setminus (A \cap X)$, the last decomposition is equivalent to the decomposition (2.2) for the condition $\text{CUI}(A \cap X, Y \cup B)$. \square

A.4.3 Proposition 6.5

Proof. Let $y \notin X \cup Ch(X) \cup An(X)$. Then clearly $\forall x \in X, x \notin Pa(y) \cup Dn(y)$. Hence from Proposition 6.1, $\text{CUI}(X, y)$. We apply Lemma 6.3 iteratively for each $y \notin X \cup Ch(X) \cup An(X)$ (note that the first argument is X for each CUI condition, so it is X in the result as well), and get the desired result. \square

A.4.4 Proposition 6.6

Proof. A CAI condition is stronger than a CUI condition, in that $\text{CAI}(x, y) \Rightarrow \text{CUI}(x, y) \wedge \text{CUI}(y, x)$. To be a CUI network, for each node x_i it must be the case that all other nodes

are CUI of it given its parents and descendants. This is obvious since x_i is CAI of all other nodes given its parents and children. \square

A.4.5 Lemma 6.7

Proof. To determine $h_{x_j}(\cdot)$, any $y \in Sc(x_j)$ needs to be determined. If $y \in \{x_i\} \cup Sc(x_i)$ we are done, if not its own scope is covered and therefore recursively determined by the assignment to $\{x_i\} \cup Sc(x_i)$. \square

A.4.6 Lemma 6.8

I first introduce two additional lemmas.

Lemma A.1. $Ch(\{x_i\} \cup Co(x_i)) \subseteq (\{x_i\} \cup Sc(x_i) \cup Co(x_i))$

Proof. Let $x_j \in \{x_i\} \cup Co(x_i)$, and $y \in Ch(x_j)$. If $x_j = x_i$ the proof is immediate because $Ch(x_i) \subseteq Sc(x_i)$. Assume $x_j \in Co(x_i)$. We know from Definition 6.4 that $Ch(x_j) \subseteq Sc(x_j) \subseteq (\{x_i\} \cup Sc(x_i) \cup Co(x_i))$, and this proves the lemma. \square

Lemma A.2. $An(\{x_i\} \cup Co(x_i)) \subseteq Co(x_i)$

Proof. Let $x_j \in An(x_i)$ (clearly $j < i$, therefore $x_j \notin Sc(x_i)$). Let $x_{j_1} \in Sc(x_j)$. Then $j_1 > j$ and there is an undirected path from x_{j_1} to x_j , not blocked by $Sc(x_j)$. If $j_1 \geq i$, then $x_{j_1} \in Sc(x_i) \cup \{x_i\}$ because it has an unblocked path to x_j (and from there to x_i). Otherwise, let $x_{j_2} \in Sc(x_{j_1})$, and apply the same argument to x_{j_2} . We continue until x_{j_k} such that $\forall x_y \in Sc(x_{j_k}), y > i$ at which point $x_y \in Sc(x_i) \cup \{x_i\}$ by the path $x_y, x_{j_k}, \dots, x_{j_1}, x_j, x_i$ and the recursion halts (note that it includes empty scopes), proving that $x_j \in Co(x_i)$.

It is left to prove that $An(Co(x_i)) \subseteq Co(x_i)$. Let $x_j \in Co(x_i), y \in An(x_j)$. Applying the first part of the proof on x_j , we get that $y \in Co(x_j)$. From Definition of $Co(x_j)$, we get $y < j$ and $\forall w \in Sc(y)$, either $w = x_j, w \in Sc(x_j)$ or $w \in Co(x_j)$. To show that $y \in Co(x_i)$, we need to prove for each of the cases that $w \in \{x_i\} \cup Sc(x_i) \cup Co(x_i)$.

1. If $w = x_j$ immediately $w \in Co(x_i)$.
2. If $w \in Sc(x_j)$, from $x_j \in Co(x_i)$ we get that either $w = x_i, w \in Sc(x_i)$ or $w \in Co(x_i)$.
3. If $w \in Co(x_j)$, we repeat the argument recursively $\forall z \in Sc(w)$. Note that z precedes w therefore the recursion will halt at some point.

\square

Lemma 6.8. Let $X = \{x_i\} \cup Co(x_i)$. From Lemma A.2, X has no external ancestors. From Lemma A.1, all external children of X are in $Sc(x_i)$. Therefore $A \setminus (X \cup An(X) \cup Ch(X)) = A \setminus (X \cup Sc(x_i))$ and the result is immediate from Proposition 6.5. \square

A.4.7 Proposition 6.9

Proof. Let x be a node in G' . Let $Y = A \setminus (x \cup \text{In}(x))$ in G and $\hat{Y} = A \setminus (x \cup \text{Pa}(x) \cup \text{Dn}(x))$ in G' . By definition of G , we know that $\text{CAI}(Y, x)$, so also $\text{CUI}(Y, x)$. Let $y \notin Y, y \neq x$ (so $y \in \text{In}(x)$ in G). If $y \in \hat{Y}$, then $y \notin \text{Pa}(x) = \text{In}(x)$ in G' . Then the arc (y, x) was removed, meaning that $y \in \text{Dn}(x)$. It therefore must be the case that $y \notin \hat{Y}$. Therefore $\hat{Y} \subseteq Y$ hence $\text{CUI}(\hat{Y}, x)$. \square

A.4.8 Proposition 6.10

Proof. For G to become a CUI tree, for each cycle at least one weak link must be removed. Since (y_i, y_{i+1}) is the only weak link for c , it must be removed. By Proposition 6.9, the variable ordering must ensure that y_{i+1} is an ancestor of y_i . This can be done through the path according to the order of p , or there might be another path from y_{i+1} to y_i . Let p_1 be such path. Then the combination of p_1 and p is another cycle c_1 , which therefore must be broken. Since p comprises of strong links, there must be at least one weak link (u, v) in p_1 . For (u, v) to be removed, v must be an ancestor of u . This can be done through the path in the cycle c_1 , and this path includes p , or through another path if such exists, for which we can repeat the argument. At each stage we get a larger cycle c_i , and a larger path $p_i \supset p_{i-1}$. Therefore at some point there will be just one path p_i that must be guaranteed by the variable ordering, and this path includes p . \square

A.4.9 Proposition 6.11

Proof. I show that Algorithm CUI-Nested leads to a functional decomposition over cliques. The outer loop in the algorithm maintains the following iteration properties:

1. $\forall a \in A_j^i, Q_j^i \subseteq \text{Pa}(a)$
2. u_j^i is defined over K_j^i
3. $A_j^i \cup x_j$ is a clique

These properties hold trivially after initialization. Assume they are valid for all factors stored in the network until outer iteration j and inner iteration i , I next show that they remain valid for each factor u_r^d that is created in iteration j, i :

1. By definition $A_r^d = A_j^i \cup x_j$. From previous iteration and definition of Q_r^d , $\forall a \in A_j^i, Q_r^d \subseteq Q_j^i \subseteq \text{Pa}(a)$. From definitions of Q_j^i and Q_r^d we get $\text{Pa}(x_j) \supseteq Q_j^i \supseteq Q_r^d$, and together it yields the result.
2. u_r^d is a factor in the CUI decomposition of $u_j^i(K_j^i)$ over G_j^i . Its scope contains: (i) the nodes that are not affected by the last CUI decomposition, i.e. in $K_j^i \setminus Q_j^i = A_j^i \cup x_j = A_r^d$, (ii) its node x_r , and (iii) the parents $\text{Pa}(x_r)$ which were not fixed

in u_j^i (i.e. $Pa(x_r \cap K_j^i)$). We know $K_j^i = A_j^i \cup x_j \cup Q_j^i$, and $x_j \notin Pa(x_r)$ (because $x_r \in Pa(x_j)$), and also $Pa(x_r) \cap A_j^i = \emptyset$ (using a similar argument and property 1). Therefore $(Pa(x_r) \cap K_j^i) \subset Q_j^i$, and from (i),(ii),(iii) we get that $K_r^d = A_r^d \cup x_r \cup Q_r^d$.

3. A_r^d is a clique by its definition and the same property of previous iteration. $x_r \in Q_j^i$, therefore from property 1 of previous iteration $x_r \in Pa(a)$ for each $a \in A_j^i$. Also $x_r \in Q_j^i \subseteq Pa(x_j)$ (the last containment is immediate from definition of Q_j^i). Therefore x_r is a parent of all members of A_r^d , and as a result $A_r^d \cup x_r$ is a clique.

By the iteration properties, either K_r^d is a clique, or Q_r^d is non empty and decomposition can be applied once we reach node r in the outer loop. At the end of the process all factors which are not defined on cliques where removed. All the factors that remained are defined on cliques. $u(A)$ can be still represented by the new set of factors since we only applied valid decompositions to its factors. \square

Appendix B

Additional Simulation Data

For the sake of completeness, I provide the results of the tests described in Section 4.6.1. In all the tests I used a GAI structure of 10 attributes with $\xi = 3$ and $e = 5$. Table B.1 summarizes the results of varying the means and variances. The general seller mean is 600 in all the tests. The results also include a second-score auction (SE-SC) (in which sellers are provided with the buyer's utility function) which is guaranteed to be within δ of the optimum given SB sellers. In addition, I provide the result of an AP auction that uses a randomly generated additive utility function (RAND)—which obviously performs much worse than AP. The results in which all auctions achieve maximal efficiency occur in the two cases where the surplus of all the auctions is zero, because the buyer's mean is above the sellers', and the variance is low. Figure B.1 shows the performance of GAI and AP as a function of the number of sellers m and the size of the attribute domain d .

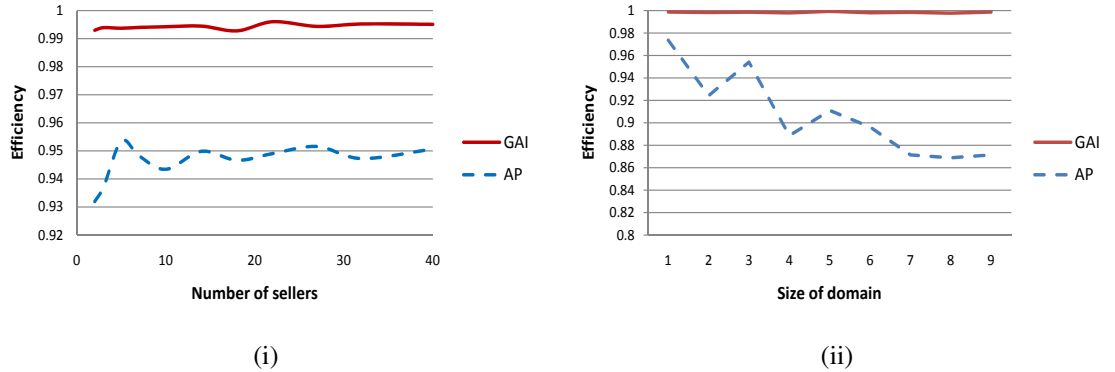


Figure B.1: Efficiency of each auction as a function of: (i) the number of sellers (m), (ii) the size of the domain of each attribute (d).

SE-SC	GAI	AP	RAND	Buyer	Seller	Variance
1	0.982	0.781	0.545	700	600	250
1	0.994	0.901	0.838	600	600	250
1	0.995	0.936	0.905	500	600	250
1	0.997	0.958	0.927	400	600	250
0.999	0.997	0.966	0.942	300	600	250
1	0.960	0.625	0.462	700	600	200
1	0.991	0.898	0.819	600	600	200
1	0.992	0.927	0.912	500	600	200
1	0.996	0.962	0.949	400	600	200
1	0.997	0.971	0.950	300	600	200
1	1	1	1	700	600	100
0.999	0.963	0.906	0.831	600	600	100
0.999	0.984	0.955	0.939	500	600	100
1	0.991	0.970	0.962	400	600	100
1	0.994	0.981	0.974	300	600	100
1	1	1	1	700	600	50
0.999	0.914	0.908	0.830	600	600	50
1	0.974	0.972	0.962	500	600	50
1	0.988	0.985	0.981	400	600	50
0.999	0.989	0.989	0.986	300	600	50

Table B.1: Efficiency achieved by each auction type, given buyer and seller means, and variance of both.

Bibliography

- Abbas, Ali and Ronald A. Howard. 2005. Attribute dominance utility, *Decision Analysis*, 2, 185–206.
- Abrams, Zoé and Michael Schwarz. 2007. Ad auction design and user experience, in *Workshop on Internet and Network Economics (WINE)*.
- Ahuja, Ravindra K., Thomas L. Magnanti, and James B. Orlin. 1993. *Network Flows*, Prentice-Hall.
- Athey, Susan and Glenn Ellison. 2007. Position auctions with consumer search. Working paper, Harvard University.
- Ausubel, Lawrence M. 2004. An efficient ascending-bid auction for multiple objects, *American Economic Review*, 94(5), 1452–1475.
- Ausubel, Lawrence M. and Paul Milgrom. 2002. Ascending auctions with package bidding, *Frontiers of Theoretical Economics*, 1, 1–42.
- Bacchus, Fahiem and Adam Grove. 1995. Graphical models for preference and utility, in *Eleventh Conference on Uncertainty in Artificial Intelligence*, Montreal, 3–10.
- Beall, Stewart, Craig Carter, Phillip L. Carter, Thomas Germer, Thomas Hendrick, Sandy Jap, Lutz Kaufmann, Debbie Maciejewski, Robert Monczka, and Ken Petersen. 2003. The role of reverse auctions in strategic sourcing, Center for Advanced Purchasing Studies (CAPS), Tempe, AZ.
- Beil, Damian R. and Lawrence M. Wein. 2003. An inverse-optimization-based auction for multiattribute RFQs, *Management Science*, 49, 1529–1545.
- Bichler, Martin. 2000. An experimental analysis of multi-attribute auctions, *Decision Support Systems*, 29, 249–268.
- Bichler, Martin. 2001. *The Future of e-Markets: Multi-Dimensional Market Mechanisms*, Cambridge University Press.
- Bichler, Martin, Andrew Davenport, Gail Hohner, and Jayant Kalagnanam. 2006. Industrial procurement auctions, in Cramton et al. (2006), chap. 23, 594–612.
- Bichler, Martin and Jayant Kalagnanam. 2005. Configurable offers and winner determination in multi-attribute auctions, *European Journal of Operational Research*, 160, 380–394.
- Börger, Tilman, Ingemar Cox, Martin Pesendorfer, and Vaclav Petricek. 2006. Equilibrium bids in sponsored search auctions: Theory and evidence. Working paper, University of Michigan.
- Boutilier, Craig, Fahiem Bacchus, and Ronen I. Brafman. 2001. UCP-networks: A directed graphical representation of conditional utilities, in *Seventeenth Conference on Uncertainty in Artificial Intelligence*, Seattle, 56–64.

- Boutilier, Craig, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. 2004a. CP-nets: A tool for representing and reasoning about conditional ceteris paribus preference statements, *Journal of Artificial Intelligence Research*, 21, 135–191.
- Boutilier, Craig and Holger H. Hoos. 2001. Bidding languages for combinatorial auctions, in *Seventeenth International Joint Conference on Artificial Intelligence*, 1211–1217.
- Boutilier, Craig, Tuomas Sandholm, and Rob Shields. 2004b. Eliciting bid taker non-price preferences in (combinatorial) auctions., in *Nineteenth National Conference on Artificial Intelligence*, San Jose, CA, 204–211.
- Brafman, Ronen I. and Carmel Domshlak. 2002. Introducing variable importance trade-offs into CP-nets, in *Eighteenth Conference on Uncertainty in Artificial Intelligence*, Edmonton, AB, 69–76.
- Brafman, Ronen I., Carmel Domshlak, and Tanya Kogan. 2004. Compact value-function representations for qualitative preferences, in *Twentieth Conference on Uncertainty in Artificial Intelligence*, Banff, BC, 51–59.
- Branco, Fernando. 1997. The design of multidimensional auctions, *RAND Journal of Economics*, 28(1), 63–81.
- Braziunas, Darius and Craig Boutilier. 2005. Local utility elicitation in GAI models, in *Twenty-first Conference on Uncertainty in Artificial Intelligence*, Edinburgh, 42–49.
- Che, Yeon-Koo. 1993. Design competition through multidimensional auctions, *RAND Journal of Economics*, 24(4), 668–680.
- Clemen, Robert T. 1996. *Making Hard Decisions: An Introduction to Decision Analysis*, Duxbury Press.
- Conen, Wolfram and Tuomas Sandholm. 2000. Preference elicitation in combinatorial auctions, in *Second ACM Conference on Electronic Commerce*, 21–29.
- Cramton, Peter, Yoav Shoham, and Richard Steinberg. 2006. *Combinatorial Auctions*, MIT Press.
- David, Esther, Rina Azoulay-Schwartz, and Sarit Kraus. 2002. An English auction protocol for multi-attribute items, in *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, Springer, vol. 2531 of *Lecture Notes in Artificial Intelligence*, 52–68.
- Debreu, Gerard. 1959. Topological methods in cardinal utility theory, in K. J. Arrow, S. Karlin, and P. Suppes, (Eds.) *Mathematical Methods in the Social Sciences*, Stanford University Press, 16–26.
- Dechter, Rina. 1996. Bucket elimination: A unifying framework for probabilistic inference, in *Twelfth Conference on Uncertainty in Artificial Intelligence*, Portland, OR, 211–219.

- Domshlak, Carmel. 2002. *Modeling and Reasoning about Preferences with CP Nets*, Ph.D. thesis, Ben-Gurion University of the Negev, Beer Sheva, Israel.
- Dyer, James S. and Rakesh K. Sarin. 1977. An axiomatization of cardinal additive conjoint measurement theory, Working Paper 265, WMSI, UCLA.
- Dyer, James S. and Rakesh K. Sarin. 1979. Measurable multiattribute value functions, *Operations Research*, 27, 810–822.
- Dyer, James S. and Rakesh K. Sarin. 1982. Relative risk aversion, *Management Science*, 28, 875–886.
- Economides, Nicholas and Robert A. Schwartz. 1995. Electronic call market trading, *Journal of Portfolio Management*, 21(3), 10–18.
- Edelman, Benjamin, Michael Ostrovsky, and Michael Schwarz. 2007. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords, *American Economic Review*, 97, 242–259.
- Engel, Yagil and David Maxwell Chickering. 2008. Incorporating user utility into sponsored-search auctions (short paper), in *Seventh International Conference on Autonomous Agents and Multiagent Systems*, 1565–1568.
- Engel, Yagil, Kevin M. Lochner, and Michael P. Wellman. 2007. Preference representation for multi-unit multiattribute auctions, in *AAAI-07 Workshop on Preference Handling in Artificial Intelligence*.
- Engel, Yagil and Michael P. Wellman. 2007. Generalized value decomposition and structured multiattribute auctions, in *Eighth ACM Conference on Electronic Commerce*, 227–236.
- Engel, Yagil and Michael P. Wellman. 2008a. CUI networks: A graphical representation for conditional utility independence, *Journal of Artificial Intelligence Research*, 31, 83–112.
- Engel, Yagil and Michael P. Wellman. 2008b. Evaluating the benefits of generalized additive representation in a multiattribute auction setting, in *Fourth Multidisciplinary Workshop on Advances in Preference Handling*.
- Engel, Yagil, Michael P. Wellman, and Kevin M. Lochner. 2006. Bid expressiveness and clearing algorithms in multiattribute double auctions, in *Seventh ACM Conference on Electronic Commerce*, Ann Arbor, MI, 110–119.
- Fink, Eugene, Josh Johnson, and Jenny Hu. 2004. Exchange market for complex goods: Theory and experiments, *Netnomics*, 6(1), 21–42.
- Fishburn, Peter C. 1967a. Interdependence and additivity in multivariate, unidimensional expected utility theory, *International Economic Review*, 8, 335–342.
- Fishburn, Peter C. 1967b. Methods of estimating additive utilities, *Management Science*, 13(7), 435–453.

- Fishburn, Peter C. 1976. Cardinal utility: An interpretive essay, *Rivista Internazionale di Scienze Economiche e Commerciali*, 23(12), 1102–1114.
- Fishburn, Peter C. 1982. *The Foundations of Expected Utility*, Reidel.
- Fisher, M. L., R. Jaikumar, and L. N. Van Wassenhove. 1986. A multiplier adjustment method for the generalized assignment problem, *Management Science*, 32(9), 1095–1103.
- Gong, Jianli. 2002. *Exchanges for Complex Commodities: Search for Optimal Matches*, Master's thesis, University of South Florida.
- Gonzales, Christophe and Patrice Perny. 2004. GAI networks for utility elicitation, in *Ninth International Conference on the Principles of Knowledge Representation and Reasoning*, Whistler, BC, 224–234.
- Gonzales, Christophe and Patrice Perny. 2005. GAI networks for decision making under certainty, in *IJCAI-05 Workshop on Advances in Preference Handling*, Edinburgh.
- Gorman, William M. 1968. The structure of utility functions, *Review of Economic Studies*, 35, 367–390.
- Guestrin, Carlos, Daphne Koller, Ronald Parr, and Shobha Venkataraman. 2003. Efficient solution algorithms for factored MDPs, *Journal of Artificial Intelligence Research*, 19, 399–468.
- Järvelin, Kalervo and Jaana Kekäläinen. 2000. IR evaluation methods for retrieving highly relevant documents, in *Proceedings of SIGIR*, 41–48.
- Kalagnanam, Jayant R., Andrew J. Davenport, and Ho S. Lee. 2001. Computational aspects of clearing continuous call double auctions with assignment constraints and indivisible demand, *Electronic Commerce Research*, 1(3), 221–238.
- Keeney, Ralph L. 1971. Utility independence and preferences for multiattributed consequences, *Operations Research*, 19(4), 875–893.
- Keeney, Ralph L. and Howard Raiffa. 1976. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Wiley.
- Koppius, Otto. 2002. *Information Architecture and Electronic Market Performance*, Ph.D. thesis, Erasmus University, Rotterdam, The Netherlands.
- Krantz, David H., R. Duncan Luce, Patrick Suppes, and Amos Tversky. 1971. *Foundations of Measurement*, vol. 1, New York: Academic Press.
- La Mura, Piero and Yoav Shoham. 1999. Expected utility networks, in *Fifteenth Conference on Uncertainty in Artificial Intelligence*, Stockholm, 366–373.
- Lahaie, Sébastien. 2006. An analysis of alternative slot auction designs for sponsored search, in *Seventh ACM Conference on Electronic Commerce*, 218–227.

- Lahaie, Sébastien and David M. Pennock. 2007. Revenue analysis of a family of ranking rules for keyword auctions, in *Eighth ACM Conference on Electronic commerce*, 50–56.
- Lehmann, Benny, Daniel Lehmann, and Noam Nisan. 2006. Combinatorial auctions with decreasing marginal utilities, *Games and Economic Behavior*, 55, 270–296.
- Lehmann, Daniel, Liadan I. O’Callaghan, and Yoav Shoham. 2002. Truth revelation in approximately efficient combinatorial auctions, *Journal of the ACM*, 49, 577–602.
- Leyton-Brown, Kevin, Yoav Shoham, and Moshe Tennenholtz. 2000. An algorithm for multi-unit combinatorial auctions, in *Seventeenth National Conference on Artificial Intelligence*, 56–61.
- Lochner, Kevin M. 2008. *Multiattribute Call Markets*, Ph.D. thesis, University of Michigan, Ann Arbor, MI.
- MacKie-Mason, Jeffrey K. and Hal R. Varian. 1994. Generalized Vickrey auctions, Tech. rep., University of Michigan.
- Mas-Colell, Andreu, Michael D. Whinston, and Jerry R. Green. 1995. *Microeconomic Theory*, New York: Oxford University Press.
- Metti, Theresa, Rob Harlan, Quentin Samelson, Tom Moore, Thomas Morris, Ron Sorensen, Avner Schneur, Olga Raskina, Rina Schneur, Joshua Kanner, Kevin Potts, and Jeffrey Robbins. 2005. Reinventing the supplier negotiation process at Motorola, *Interfaces*, 35(1), 7–23.
- Myerson, Roger B. 1981. Optimal auction design, *Mathematics of Operations Research*, 6(1), 58–73.
- Myerson, Roger B. and Mark A. Satterthwaite. 1983. Efficient mechanisms for bilateral trading, *Journal of Economic Theory*, 28, 265–281.
- Nilsson, D. 1998. An efficient algorithm for finding the M most probable configurations in probabilistic expert systems, *Statistics and Computing*, 8(2), 159–173.
- Nisan, Noam. 2000. Bidding and allocation in combinatorial auctions, in *Second ACM Conference on Electronic Commerce*, Minneapolis, MN, 1–12.
- Parkes, David. 2001. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*, Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.
- Parkes, David C. and Jayant Kalagnanam. 2005. Models for iterative multiattribute procurement auctions, *Management Science*, 51, 435–451.
- Pearl, Judea. 1988. *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann.
- Pearl, Judea and Azaria Paz. 1987. Graphoids: A graph based logic for reasoning about relevance relations, in B. Du Boulay, David Hogg, and Luc Steels, (Eds.) *Advances in Artificial Intelligence II*, North Holland, 357–363.

- Ronen, Amir and Daniel J. Lehmann. 2005. Nearly optimal multi attribute auctions, in *Sixth ACM Conference on Electronic Commerce*, 279–285.
- Rothkopf, Michael H., Thomas J. Teisberg, and Edward P. Kahn. 1990. Why are vickrey auctions rare?, *Journal of Political Economy*, 98(1), 94–109.
- Saaty, Thomas L. 1980. *The Analytic Hierarchy Process*, McGraw Hill.
- Sandholm, Tuomas. 2002. Algorithm for optimal winner determination in combinatorial auctions, *Artificial Intelligence*, 135, 1–54.
- Sandholm, Tuomas. 2007. Expressive commerce and its application to sourcing: How we conducted \$35 billion of generalized combinatorial auctions, *AI Magazine*, 28(3), 45–58.
- Sandholm, Tuomas and Craig Boutilier. 2006. Preference elicitation in combinatorial auctions, in Cramton et al. (2006), chap. 10, 233–263.
- Sandholm, Tuomas, David Levine, Michael Concordia, Paul Martyn, Rick Hughes, Jim Jacobs, and Dennis Begg. 2006. Changing the game in strategic sourcing at Procter & Gamble: Expressive competition enabled by optimization, *Interfaces*, 36(1), 55–68.
- Sandholm, Tuomas and Subhash Suri. 2001. Side constraints and non-price attributes in markets, in *IJCAI-01 Workshop on Distributed Constraint Reasoning*, Seattle, WA.
- Sandholm, Tuomas, Subhash Suri, Andrew Gilpin, and David Levine. 2005. CABOB: A fast optimal algorithm for winner determination in combinatorial auctions, *Management Science*, 51, 373–390.
- Schvartzman, L. Julian and Michael P. Wellman. 2007. Market-based allocation with indivisible bids, *Production and Operations Management*, 16(4), 495–509.
- Shachat, Jason and J. Todd Swarthout. 2002. Procurement auctions for differentiated goods, IBM Research Report RC22587, IBM T.J. Watson Research Laboratory.
- Shoham, Yoav. 1997. A symmetric view of probabilities and utilities, in *Fifteenth International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1324–1329.
- Strecker, Stefan. 2003. Preference revelation in multi-attribute reverse English auctions: A laboratory study, in *Twenty-Fourth International Conference on Information Systems*, 271–282.
- Strecker, Stefan and Stefan Seifert. 2004. Electronic sourcing with multi-attribute auctions, in *37th Annual Hawaii International Conference on System Sciences*, 701–712.
- Sunderam, Aditya V. and David C. Parkes. 2003. Preference elicitation in proxied multiattribute auctions., in *Fourth ACM Conference on Electronic Commerce*, San Diego, 214–215.
- Tatman, Joseph A. and Ross D. Shachter. 1990. Dynamic programming and influence diagrams, *IEEE Transactions on Systems, Man and Cybernetics*, 20, 365–379.

- Thiel, Stuart E. 1988. Multidimensional auctions, *Economic Letters*, 28, 37–40.
- Varian, Hal. 2007. Position auctions, *International Journal of Industrial Organization*, 25, 1163–1178.
- von Neumann, John and Oskar Morgenstern. 1944. *Theory of Games and Economic Behavior*, Princeton University Press.
- Von Stengel, Bernhard. 1988. Decomposition of multiattribute expected-utility functions, *Annals of Operations Research*, 16, 161–184.
- Von Winterfeldt, Detlof and Ward Edwards. 1986. *Decision Analysis and Behavioral Research*, Cambridge University Press.
- Vulkan, Nir and Nicholas R. Jennings. 2000. Efficient mechanisms for the supply of services in multi-agent environments, *Decision Support Systems*, 28, 5–19.
- Wellman, Michael P. and Jon Doyle. 1992. Modular utility representation for decision-theoretic planning, in *First International Conference on Artificial Intelligence Planning Systems*, College Park, MD, 236–242.
- Wellman, Michael P., Amy Greenwald, and Peter Stone. 2007. *Autonomous Bidding Agents: Strategies and Lessons from the Trading Agent Competition*, MIT Press.
- Wellman, Michael P., Anna Osepayshvili, Jeffrey K. MacKie-Mason, and Daniel M. Reeves. 2008. Bidding strategies for simultaneous ascending auctions, Tech. rep., University of Michigan.
- Wurman, Peter R. and Michael P. Wellman. 2000. AkBA: A progressive, anonymous-price combinatorial auction, in *Second ACM Conference on Electronic Commerce*, 21–29.
- Wurman, Peter R., Michael P. Wellman, and William E. Walsh. 2001. A parametrization of the auction design space, *Games and Economic Behavior*, 35, 304–338.